NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A PROTOTYPE OF PILOT KNOWLEDGE
EVALUATION BY AN INTELLIGENT CAI
SYSTEM USING A BAYESIAN DIAGNOSTIC MODEL

by

Yodchai Rugsumruad

June 1987

Thesis Advisor: Neil C. Rowe

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION unclassified	1b RESTRICTIVE MARKINGS			
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.			
2b DECLASSIFICATION/DOWNGRADING SCHEDULE				
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)			

6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School	6b OFFICE SYMBOL (If applicable) 52	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000		7b ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000
8a NAME OF FUNDING/SPONSORING ORGANIZATION	8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS			
	PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification) A PROTOTYPE OF PILOT KNOWLEDGE EVALUATION BY AN INTELLIGENT CAI SYSTEM USING A BAYESIAN DIAGNOSTIC MODEL

12 PERSONAL AUTHOR(S) Rugsumruad, Yodchai

13a TYPE OF REPORT Master's Thesis	13b TIME COVERED FROM _____ TO _____	14 DATE OF REPORT (Year, Month, Day) 1987 June	15 PAGE COUNT 77

16 SUPPLEMENTARY NOTATION

17	COSATI CODES		18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Computer-Aided Instruction; CAI System; Intelligent CAI System

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

This thesis describes the Testing system, a CAI system which monitors the progress of a student and automatically adjusts the lesson to suit his/her understanding of the material being presented. Each lesson is divided into three segments. The first segment consists of questions critical to the basic understanding of the lesson, the second questions complementary to the lesson, and the third questions supplementary. After the first segment, the student's performance is evaluated and a decision is made whether or not he/she should continue on to the complementary segment of the lesson or go on the the next lesson. Similarly, in the second segment, a decision whether to continue on to the supplementary segment or to the next lesson is made. Thus, a proficient student can progress through the material quickly, a slow learner slowly. The questions are grouped into types, and each of them is ranked as an easy or

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION unclassified	
22a NAME OF RESPONSIBLE INDIVIDUAL Prof. Neil C. Rowe	22b TELEPHONE (Include Area Code) (408) 646-2462	22c OFFICE SYMBOL Code 52Rp

DD FORM 1473, 84 MAR   83 APR edition may be used until exhausted   SECURITY CLASSIFICATION OF THIS PAGE
All other editions are obsolete

BLOCK 19 CONTINUATION

difficult concept to provide more precise guidance for the
student's improvement. An example lesson is provided as an
illustration.

A Prototype Of Pilot Knowledge Evaluation by An Intelligent CAI
System Using a Bayesian Diagnostic Model

by

Yodchai Rugsumruad
Commander , Royal Thai Navy
B.S., Royal Thai Naval Academy, 1972

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
June 1987

ABSTRACT

This thesis describes the Testing system, a CAI system which monitors the progress of a student and automatically adjusts the lesson to suit his/her understanding of the material being presented. Each lesson is divided into three segments. The first segment consists of questions critical to the basic understanding of the lesson, the second questions complementary to the lesson, and the third questions supplementary. After the first segment, the student's performance is evaluated and a decision is made whether or not he/she should continue on to the complementary segment of the lesson or go on to the next lesson. Similarly, in the second segment, a decision whether to continue on to the supplementary segment or to the next lesson is made. Thus, a proficient student can progress through the material quickly, a slow learner slowly. The questions are grouped into types, and each of them is ranked as an easy or difficult concept to provide more precise guidance for the student's improvement. An example lesson is provided as an illustration.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

Over the years, the computer has become an indispensable tool in all walks of life. One of its important applications has been in automated instruction. It has led to a field of research called Computer-Aided Instruction (CAI). CAI is a powerful methodology for enhancing the overall quality and effectiveness of education. Many CAI systems have been developed for a variety of knowledge domains. Experience has shown that such systems often cause annoyance to fast learners and, at the other extreme, discouragement to the slow ones. Moreover, since learning is dynamic and non-uniform, it is desirable for an instructional and diagnostic system to adjust itself to the student. This helps accommodate the student's familiarity and talents for different topics so that an intelligent/knowledgeable student is not surfeited with redundant information, nor a student who has extreme difficulties in a topic be dampened of motivation.

This thesis proposes a CAI system which includes a diagnostic model to test how well the learner has absorbed the instructional material and diagnose precisely where the learner has difficulties. The diagnostic model being implemented here was first developed by Professor Taracad R. Sivasankaran and Professor Tung Bui [Ref. 1] and based on the Bayesian Analysis.

The proposed system for Computer-Aided Instruction separates the questions into three segments. The first segment consists of questions which are critical to the understanding of the lesson. The second and the third segments consists of questions which are complementary and supplementary to the critical segment. By separating the questions in this manner, the system can determine after the learner has completed the critical questions whether or not the student has mastered the material. The system determines the student's knowledge by examining the number of correctly answered questions and uses this information to calculate the (posterior) probability of the student's ability to answer the complementary and supplementary questions. If this probability exceeds the upper cut-off criterion, the student is assumed to have mastered the lesson and may continue on to the next lesson. However, if the probability is below the lower cut-off criterion, the student is deemed to have insufficient knowledge about

8

the lesson and the system will not allow the student to continue with lesson. If the probability is in between the two cut-off criteria, the student's knowledge of the lesson cannot be determined by the student's performance; thus, further testing/questioning is required. The system will then present the student with a question from the complementary segment. After the student completed the test question, the same probabilistic analysis is applied and a decision is made as to whether or not the student should answer the supplementary questions or go on to the next lesson.

Chapter 2 provides the necessary background concerning the general description of Computer-Aided Instruction, the basics of the Bayesian Analysis, and demonstrates the design of the questions for the Testing System. For illustration purposes, the questions are drawn from the "Student Pilot's Flight Manual" [Ref. 2]. Chapter 3 presents the diagnostic model. Chapter 4 describes the system architecture, data structures, the detailed design and descriptions of the program. Finally, Chapter 5 concludes the thesis and suggests future directions.

II. BACKGROUND

A. COMPUTER-AIDED INSTRUCTION

1. Definition

The term CAI (Computer-Aided Instruction) is commonly used to refer to all aspects of computers in education. The definition of CAI has been narrowed and considered, a distinct subject within a broader category termed Computer Based Education (CBE). While the actual boundaries of the definition of CAI are still somewhat blurred, a general consensus of what constitutes a CAI has evolved:

"CAI is a term used to described the process of interactive teaching with the aid of a computer. The process directly involves the computer in the storage and presentation of instructional materials to the student. Both written material and graphics are presented to the student in a logical manner". [Ref. 3]

This definition, while broad, serves to distinguish CAI from another growing field of computers in education termed Computer Managed Instruction (CMI). In CMI, the computer is not normally used as a direct interface with the student to effect learning. Instead, the computer is used as a management tool to assist the instructor. The materials presented to the student in the CMI model are not of a direct educational nature and include such things as tests and management of the flow of a student's education. The computer is used as a gate to control student progress and to provide extensive managerial data to the instructor to enable more individual tailoring of instruction.

Despite the fact that a few current authors continue to refer to both CAI and CMI as a single entity. There are two important distinctions between CAI and CMI:

- CAI is usually performed in an on-line real time or near real time basis in order to provide immediate feedback of an instructional nature to the student. CMI, on the other hand, is typically not a real time process and focuses on the batch processing of data for later presentations to the instructor. [Ref. 4]

- CAI is more intensive both in hardware and software [Ref. 5]. In order to implement a successful CAI system, students must have ready access to computer terminal, necessitating a significant investment in hardware. In addition, the software (which the company refers to as courseware) required to effect a CAI is complex and long lead times and high development costs are the rule. In contrast, CMI can be effective with only a few terminals for joint instructor and student use. Software development for CMI tends to be relatively straight forward and relatively inexpensive.

2. Basic Categories of CAI

CAI can be further subdivided into five teaching strategies: (1) Drill and practice; (2) Tutorial; (3) Problem solving tasks; (4) Simulation; and (5) Instructional games [Ref. 6].

a. Drill-and-Practice

The user is assumed to have already learned the basis of the material to be presented through traditional classroom lecture techniques. The goal of computer is to cement the retention of the material by repeated practice and exercise. This method provides the student with the capability to work through the large array of problems and to develop an intuitive understanding of the procedures used. [Ref. 7]

Drill-and-Practice programs tend to be organized linearly with a brief introduction followed by the presentation of a series of problems or solution step. A series of preselected problems can be provided from outside material by the student through appropriate interfacing with the program. The latter method is more flexible and provides for more student involvement. The embedding of the actual problems in the software significantly restricts the scope of the program and requires extensive maintainance to change the problem sets. Drill-and-Practice type software is particularly well suited to courses where a repetitive practice is required to learn a skill or a concept. The linear sequencing of solution steps for many mathematics and statistical problems can easily be accommodated using the Drill-and-Practice strategy.

b. Tutorial

The tutorial strategy encompasses a great deal more than the relatively simple Drill-and-Practice strategy. Whereas Drill-and-Practice programs serve as an adjunct to the instructor, tutorial programs strive to replace the instructor as much as possible. Logical dialogue between the student and the program about specific problems is the hallmark of this strategy. These programs are typically capable of detecting not just simple calculations and data input errors, but also errors in reasoning by the student. Extensive feedback is provided to the student to assist in correcting these errors. [Ref. 8]

The tutorial strategy is most often viewed by the layman as the epitome of what a CAI should be. The vast majority of the early large scale endeavors in the field of CAI were of this type. As computer technology has advanced, improvement have been incorporated into the tutorial strategy and it remains the most preferred of CAI methods. This technique can be tailored to fit most any learning situation but it excels in imparting complicated conceptual skills. [Ref. 9]

11

c. *Problem Solving Tasks*

This strategy encompasses programs which are written to solve certain selected problems in a specific area. This is the least well defined of the five strategies presented in that a wide variance of the sophistication can exist between different programs within this strategy. This strategy places no boundaries on the various applications or level of difficulty of problems. In general, these programs tend to be more solution oriented and less concerned with developing computational or conceptual skills of the student. Most applications within this strategy parallel the use of a simple calculator; however, in this case, the programs have the ability to solve problems of far greater complexity. This strategy is well suited to quantitative courses which are primarily interested in the final solution and not the methods or techniques to derive the solutions. [Ref. 10]

d. *Simulation*

This strategy is generally used for the generation and manipulation of data or the repetitive cycling of a model when environmental factors preclude normal methods. These environmental factors include limitations of time, money or equipments; and various safety considerations.

This strategy is based on the mathematical manipulation of complex modules which places the student in a controlled real life situation. Most programs of this type allow the user to input a series of parameters and process these, in a compressed fashion, through the module. The resultant information is displayed for user perusal and may include pertinent comments to key the student to significant data. In general, no learning path is predefined. Instead, the student is allowed to learn through actual manipulation of these processes. These types of program are frequently used in statistical analysis and can be very effective learning tools. Concepts learned via this strategy are likely to be retained by the student longer. [Ref. 11]

e. *Instructional Games*

The distinctive feature of this strategy is an attempt to use a student's competitive nature to achieve certain learning goals or skills. These programs most often pit the student against the computer in a win-lose situation in an effort to hold the student's attention and interest. Creative programming skills are required to make this strategy effective as a learning tool. This technique is ideally suited to elementary level education.

This thesis demonstrates Drill-and-Practice strategy. The Testing program is organized linearly with a brief introduction followed by the presentation of a series of questions in a domain of knowledge (field of aviation). The Testing makes use of Bayes' Theorem to adjust a lesson being presented to suit his/her understanding of the lesson.

B.   BAYESIAN ANALYSIS

Bayesian analysis is concerned with the basic problem of assessing some underlying "state of nature" that is in some way uncertain. On the basis of whatever evidence does exist, some action or actions are to be chosen from among various possible alternatives. In some cases, the task of the investigator will be that of statistical inference, the only action to be taken being the acceptance or rejection of some hypothesis. In other cases, the actions will be the implementation of various possible decisions such as that the outcomes of these decisions depend upon the unknown state of nature. [Ref. 12]

The basis for Bayesian Analysis is the *Bayes' Theorem.* this theorem provides a very powerful tool for statistical inference, especially when pooling informations from different source is appropriate. Thus, prior information resulting from economic theory and/or from previous (real or hypothetical) samples can be combined with the information embodied in new observation, and this operation can be performed formally, within a rigorous mathematical framework. [Ref. 13]

Bayes' Theorem can be explained in simple terms as follows. Assume that there are two events A and B whose joint probability density function is given $P(A,B)$. Then

$$P(A/B) = \frac{P(B/A)\,P(A)}{P(B/A)P(A) + P(B/A')P(A')} \qquad (\text{eqn } 2.1)$$

where A' represents the complement of the event A (that is, "not A") [Ref. 14].

Notice that in the above equation there are two inputs under the form of probability density. The first input is the marginal probability density p(A) which defines the knowledge of event A prior to an observation. The second input is the probability density of the event B conditional upon the occurrence of event A, i.e., the probability that event B will occur given that event A has already occurred. This latter probability density provides an additional information of the event A in the form of

conditional probability density for event A under the assumption that event B had occurred, and the above equation provides a way in which to adjust the probability of the occurrence of event A in a rigorous manner.

III. A BAYESIAN DIAGNOSTIC MODEL

A. ASSUMPTIONS

Assuming that the student has completed the study of the domain material and is ready to be tested upon his level of assimilation. At any point during a diagnostic session, the probability that the student will answer the remaining queries correctly depends on the proportion of the past queries on the same topic answered correctly.

We further assume that the domain knowledge being imparted to the student is decomposable into blocks of concepts/ideas, so that levels of knowledge about various blocks of the domain can be independently measured. Blocks can be based on chapters outlined in textbooks.

B. THE DIAGNOSTIC MODEL

The proposed diagnostic model is illustrated in Figure 3.1. The model:

1. Formulates hypotheses about a student's strengths and weaknesses regard to various concepts/ideas in a specific block of knowledge domain.

2. Tests them using a set of carefully compiled questions.

3. Derives conclusions about the student's problem areas.

It has six steps to complete the above three functions. First, the hypothesis that the student is more likely an average student than either very bright or very poor is set. In the second step, a block of concepts/ideas is selected out of the domain that the learner is to be tested upon. A series of critical questions related to that block of concepts is then presented to the student. In the fourth step, as explained later in C, depending on the number of queries correctly answered, the hypothesis can either be rejected or not. If the hypothesis is not rejected, the process of querying the student with additional questions about the same block of concepts is repeated. On the other hand, if the hypothesis is rejected, we move to the next block for either of two reasons: (1) the student is far brighter, or (2) the student is far poorer than the average student. This avoids irritating the bright student by asking too many questions and at the same time avoids discouraging the poor student as well. At the end of the session, appropriate conclusions are made about the degree of proficiency the student, his strengths, weaknesses, misconceptions, and remedial strategies are suggested.
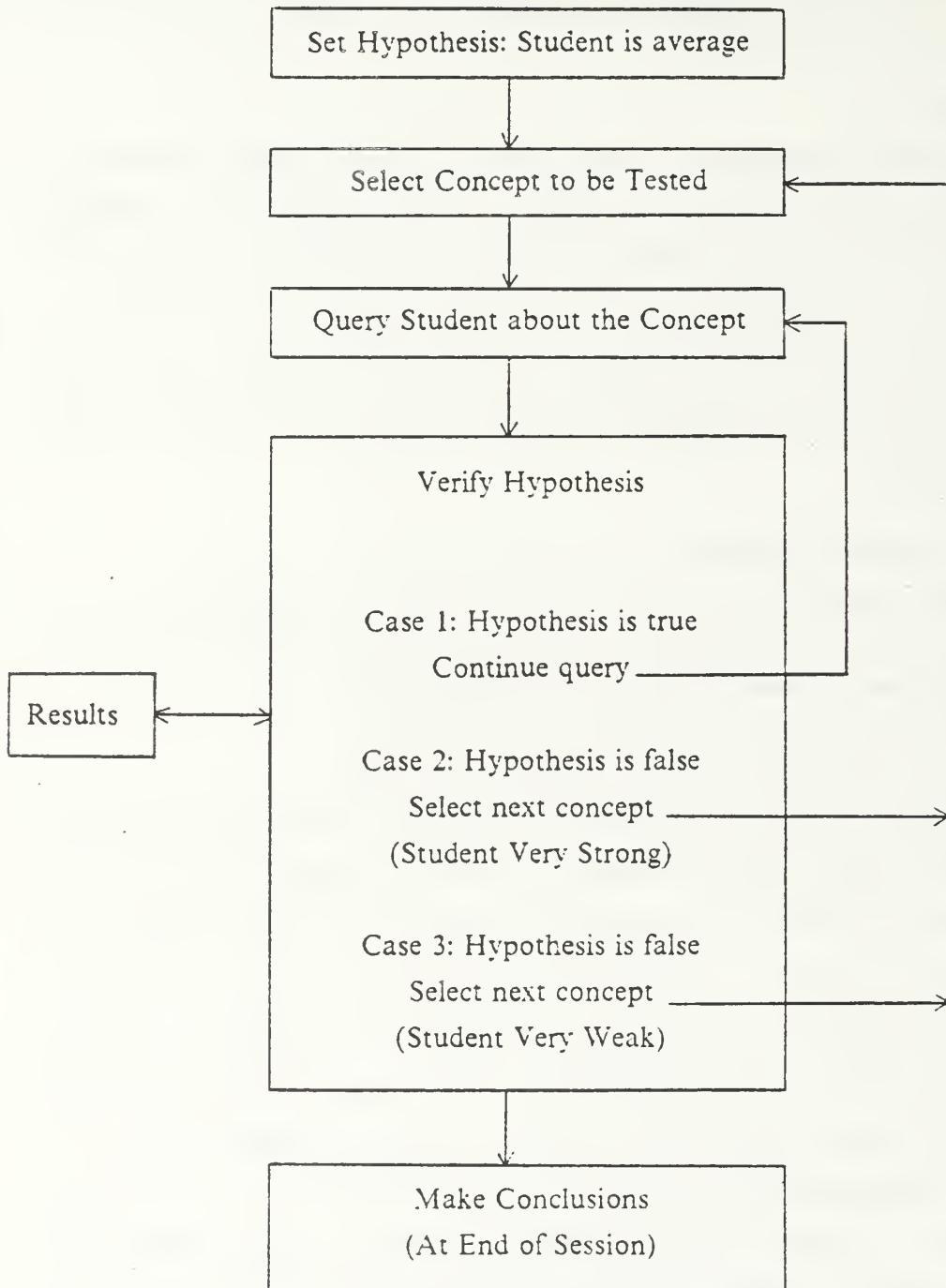
Figure 3.1   The Diagnostic Model

The power of the model lies in its ability to adapt a diagnostic session to the level of knowledge of a student automatically by limiting the type and numbers of questions. This is shown by the loops connecting step 5 to step 2 and step 3. If the initial part of a session strongly suggests that the student is knowledgeable about a certain concept, then the model skips the remaining part of the session and moves on to the next concept noting its conclusions on a global variable. The global variable forms a dynamic workspace and acts as a scratchpad. It enables the diagnostic model to create and store intermediate observations about the student performance. On the other hand, if the student is average, the entire session is offered. When all the concepts are exhausted, the Results component (global variable) is consulted and detailed diagnostic about the student's strengths, weaknesses, misconceptions, etc., is made and prescriptive measures suggested. [Ref. 15]

C. THE BAYESIAN APPROACH TO DIAGNOSTIC MODELING

It was noted that the first and fourth steps of the diagnostic model comprised of formulating and testing hypothesis about the student's level of knowledge. Since there are looping between step4 and step3, step5 and step2, and step6 and step2, depending on the proficiency of the student, it is possible that the formulation and verification steps will be executed several times sequentially. While a student is tested upon a concept using short independent questions, it becomes necessary to update the hypothesis at the end of each cycle through the loop. We consider the Bayesian statistical techniques to be appropriate and viable for such sequential updating of hypothesis, since they can based on the set of the latest finding during a session, thereby exploiting the techniques as a means of operationalizing the diagnostic model.

Before one can use the Bayesian approach, it is necessary to impart a few structural characteristics to the domain being tested upon the student. To achieve this, we shall proceed as follows. Let there be *N blocks of concepts* in the knowledge domain that are to be tested upon the student. Focus on the first block, say C_1 . Let there be at most *M questions* related to C_1 that may be posed to the student to test his expertise about the block. We shall divide the M questions into three groups, the first containing M_1 *critical questions*, the second containing M_2 *complementary questions*, and the third containing M_3 *supplementary questions*. *The critical questions* are those which are advanced in nature and cover all the important aspects of the block thoroughly (question 1 through 14 in Appendix A). In any diagnostic session with a student all the critical questions are always presented. If the student answers all of

them correctly, we presume the student need not to be subjected to the remaining (M-M_1) questions and is good enough to be tested upon the next block of concepts. *The complementary questions* are those which cover the non-critical yet important aspects of the block (question 15 through 30 in Appendix A). We presume that a close to perfect score in the critical questions combined with a perfect score in the complementary questions is once again an indicator of sufficient mastery over the block under consideration. *The supplementary questions* form the set of least advanced questions about the block (question 30 through 45 in Appendix A). They are repetitious in nature in that they do not cover any new aspects about the block. They merely split the critical and complementary questions into its elementary parts easy enough for the student to handle. The supplementary questions are meant to be posed to the student only if the performance in the critical and complementary sections has been poor enough to warrant them.

In introducing the Bayesian analysis into the above domain structure, we need to make assumptions about the prior probabilities that form the first set of estimates about the level of knowledge of a student in the knowledge domain, and a first guess of our initial degree of conviction about each of these assumptions.

The following two strategies are suggested to generate the prior probabilities. In the first, we presume that there is no real way one can estimate the level of knowledge of a student beforehand and hence we postulate that the chance of the student being proficient or ignorant is fifty-fifty. Equal probability reflects complete uncertainty. We will refer to this hereafter as a *fifty-fifty assumption*. We shall denote this prior probability by $P_u = 0.50$. The subscript u is chosen to indicate an *uninformed opinion*. In the second strategy, we assume that one way to estimate the level of knowledge of a student is to seek an estimate from an expert human instructor in the knowledge domain. Alternatively, it might also be determined by examining how the students have performed in similar tests historically. We shall denote this prior probability by using P_x. The subscript x is chosen to indicate an *expert opinion*. Historically, if students have secured a grade B average in a particular knowledge domain, one can attribute a starting probability value of say, 0.80 for P_x. Further, we assume that our degree of conviction in the uninformed versus the expert opinion is a to b where a and b are weights. We can express this in probability notations as $P(B_u) = a/(a+b)$, and $P(B_x) = b/a+b$, where $P(B_u)$ denotes the strength of belief in the uninformed opinion and $P(B_x)$ that in the expert opinion.

Let A_1 denotes the event of begining a diagnostic session by asking the student M_1 questions about C_1 and the number of questions correctly answered be R_1 Before continuing the session with the remaining $(M-M_1)$ questions, we first revise our earlier opinions about the expert (i.e., $P(B_x)$) in the light of the new evidence gathered from the diagnostic session so far (A_1) The mathematical formulation is shown below.

$$P(A_1/B_u) = \binom{M_1}{R_1} (P_u)^{R_1} (1-P_u)^{M_1-R_1} \qquad \text{(eqn 3.1)}$$

$$P(A_1/B_x) = \binom{M_1}{R_1} (P_x)^{R_1} (1-P_x)^{M_1-R_1} \qquad \text{(eqn 3.2)}$$

Applying Bayes Theorem,
$P(B_u/A_1)$ and $P(B_x/A_1)$ can be derived as follows:

$$P(B_u/A_1) = \frac{P(B_u)P(A_1/B_u)}{P(B_u)P(A_1/B_u) + P(B_x)P(A_1/B_x)} \qquad \text{(eqn 3.3)}$$

and

$$P(B_x/A_1) = 1 - P(B_u/A_1) \qquad \text{(eqn 3.4)}$$

$P(B_u/A_1)$ and $P(B_x/A_1)$ represent the revised degree of confidence in the initial hypotheses given that the student had correctly answered R_1 out of M_1 questions.

In the light of the new findings, there are two possible options. One is to use a cut-off value for $P(B_x/A_1)$, say *alpha*. Alpha is the minimum strength of belief in the expert opinion that is required to conclude that the student's ability to do a test is good. Thus, if $P(B_x/A_1) >=$ alpha, we may conclude that the student is knowledgeable enough in C_1 and not pose the remaining $M-M_1$ questions. We can avoid exasperating a good student by asking far too many questions in a field he

appears to be well informed. We can store our conclusion about the student in the Results component and proceed with the remaining blocks of concepts in the domain in the same fashion.

Conversely, if the cut-off criterion $P(B_x/A_1) \geq$ alpha for a good student is not met, it leaves us with two other possible inferences about the student. One is that the student is just an average student as was anticipated under one of the assumption; the other possibility is that the student is far poorer. We may want to make a conclusion about which of these categories the student belongs before proceeding with the $M-M_1$ questions. In fact, we may choose to spare a poor student from these questions to not turn the enthusiasm off prematurely. To distinguish the poor student from the average one, we propose using a second cut-off criterion $P(B_x/A_1) \leq$ beta. *Beta* is the minimum strength of belief in the expert opinion required to discern poor students. Thus, if $P(B_x/A_1)$ turns out to be below beta, one might skip the remaining $(M-M_1)$ questions and move to next block. The conclusion that the student is poor about the concept block is stored in the blackboard for future reference.

The other opinion arises if $P(B_x/A_1)$ falls within both cut-off criterion. In this case, we continue questioning with the set of M_2 complementary questions and repeat the procedure. If the revised probability does not meet either criterion again, the session may be continued with the set of M_3 questions. On the completion of a student session, by assessing the remarks stored on the Results component from time to time during the session, detailed diagnosis can be made about the level of proficiency of the student and intimated to him along with remedial suggestions. [Ref. 16]

D.   EXAMPLE OF BRIGHT-STUDENT PERFORMANCE

Let us assume that after a detailed analysis of the knowledge domain. an expert instructor in the domain came up with 25 (i.e., M = 25) questions about the various aspects of the block. Let these 25 questions consist of 5 critical (M_1 = 5), 8 complementary (M_2 = 8), and 12 supplementary (M_3 = 12) questions. Let two of the starting assumptions be: (1) the odds are even a student is knowledgeable or ignorant about the block of domain, or P_u = 0.50; (2) since historically students in the domain have done better, it is more likely $P_x > P_u$, hence say P_x = 0.70. Further, we shall assume that our level of confidence in these two assumptions (p_u and p_x) is 1 to 4. We can express this in probability notations as, $P(B_u)$ = 0.20 and, $P(B_x)$ = 0.80. We shall further assume the critical cut criteria, alpha and beta are 0.90 and 0.40 respectively. The following scenario demonstrates some practical implications of the proposed diagnostic model.

Suppose that when the student was asked the critical set of 5 questions, he answered 4 correctly. Let A_1 denotes this event. We have now to make a decision as to whether the student is bright or poor enough based on this performance and whether or not we should continue with the remaining 20 questions at all. To do this, we first compute $P(A_1/B_u)$ and $P(A_1/B_x)$ using equations 3.1 and 3.2 .

$$P(A_1/B_u) = \binom{5}{4} (.5)^4 (1-.5)^{5-4} = 0.156 \qquad \text{(eqn 3.5)}$$

$$P(A_1/B_x) = \binom{5}{4} (.7)^4 (1-.7)^{5-4} = 0.360 \qquad \text{(eqn 3.6)}$$

From our initial assumption about the strength of belief in the uninformed and expert opinions, we have

$$P(B_u) = 0.20 \qquad \text{(eqn 3.7)}$$

$$P(B_x) = 0.80 \qquad \text{(eqn 3.8)}$$

Applying the results of (3.5), (3.6), (3.7), and (3.8) in equations (3.3) and (3.4), we get $P(B_u/A_1) = 0.098$ and $P(B_x/A_1) = 0.902$.

Comparing $P(B_x/A_1) = 0.902$, with the criterion alpha = 0.90, we reject the assumption that the student is average. One can also note that our level of confidence between $P(B_u)$ and $P(B_x)$ has improved from 1 to 4 to 1 to 9.20 in the light of the session with the students. The conclusion here is that the student is above average. In this case we suspend asking further questions about the block of concept in consideration to the student, and proceed directly to the next block of concepts.

The fact that student was well knowledgeable about a block of domain is also noted in the Results component so that at the end of the entire diagnostic session, consolidated comments about the overall level of proficiency of the student in the knowledge domain may be made.

E. DEVELOPMENT OF QUESTIONS FOR THE TESTING SYSTEM

In order to demonstrate how the proposed system of this thesis works, the questions used in this study are drawn from "Student Pilot's Flight Manual" [Ref. 17]. This book is arranged into five chapters; (1) Part1- Before The Flight; (2) Part2- Pre-Solo; (3) Part3- Post-Solo Maneuvers; (4) Part4- Cross-Country and Night Flying; (5) Part5- The Written and Flight Test [Ref. 2: p.5]. For purposes of testing the utility of the system with question sets, only "Part1- Before The Flight" was utilized and divided into seven ideas; (1) Starting to fly; (2) The Airplane and How It Flies; (3) Cockpit-Instrument and Systems; (4) Preflight-Check; (5) Starting the Airplane; (6) Taxiing; and (7) Pre-Take-off and Cockpit Check. It should be noted that the other questions sets could be easily developed from other chapters or other texts for basic and advanced flight training. The Testing System demonstrated here may also be applied to other subject areas in which self-testing and self-evaluation is desirable.

Forty-five questions are extracted from the above text book. According to the Bayesian Diagnostic model explained in Chapter 3 of this document, these questions are separated into three sets. The first set consists of 14 questions which are considered critical to the understanding of the chapter. The second and the third sets consist of 16 complementary questions and 15 supplementary questions respectively. The first and the second sets of questions cover all important aspects about the chapter and the third set of questions does not cover new aspects about the chapter but reinforces already introduced concepts. However, the proposed model explains this concept in more details.

The forty-five questions are grouped into three types, procedural, calculative, and physical questions. Each question is ranked as an easy or difficult concept, which can be determined by the system. The purpose of this technique permits the Testing System to analyze in detail the specific level of the student's difficulty in each testing area, and provides more precise guidance for student improvement. The questions are presented in Appendix A.

IV. TESTING: A PROTOTYPE OF PILOT KNOWLEDGE EVALUATION

A. TESTING SYSTEM ARCHITECTURE

The Testing System is a CAI tutorial system with a diagnostic monitor that makes use of the Bayesian model described in Chapter 3. Testing is implemented in the knowledge domain that forms part of study in the field of aviation. The architecture of Testing is shown in Figure 4.1. The Testing System consists of four components: (1) Inference Engine; (2) Data Base; (3) Temporary Results; (4) Historical Data.
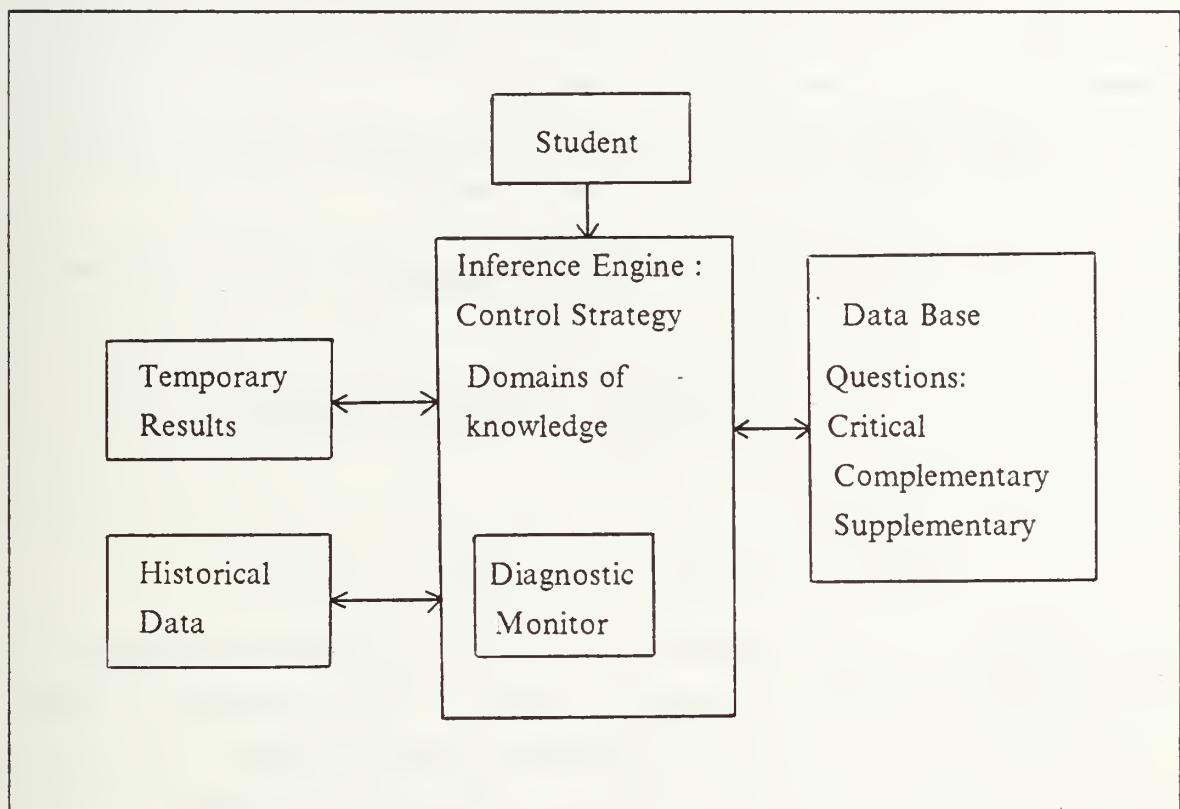
Figure 4.1   Testing System Architecture

The Inference Engine holds information about the concepts in a domain of knowledge(field of aviation). It also contains the control strategies to access the other components in the system as well as the student user. Above all, it accommodates the Diagnostic Monitor that contains the proposed Bayesian Model.

23

The Data Base consists of a bank of critical, complementary, and supplementary questions on the various concepts represented in the domain of knowledge (field of aviation) for this prototype. Examples of such questions are shown in Appendix A.

The Temporary Results component stores results after each question and essentially acts as a scratch-pad. The results are discarded once you exit the program. The Historical Data component holds any data about the student's performance in each test that are needed to be kept for future use. With this component, Testing can determine if the student's performance is better or worse than the previous one.

B.   DATA STRUCTURES

For efficient operation the program should not be spending a lot of time to search for the data needed to be presented to the student. The required questions need to reside in memory in a data structure that allow quick access. The method by which books/concepts and ideas are numbered is also important.

In this thesis, a logical tree structure was used to store the questions bank as shown in Figure 4.2. It is implemented with the use of menus.
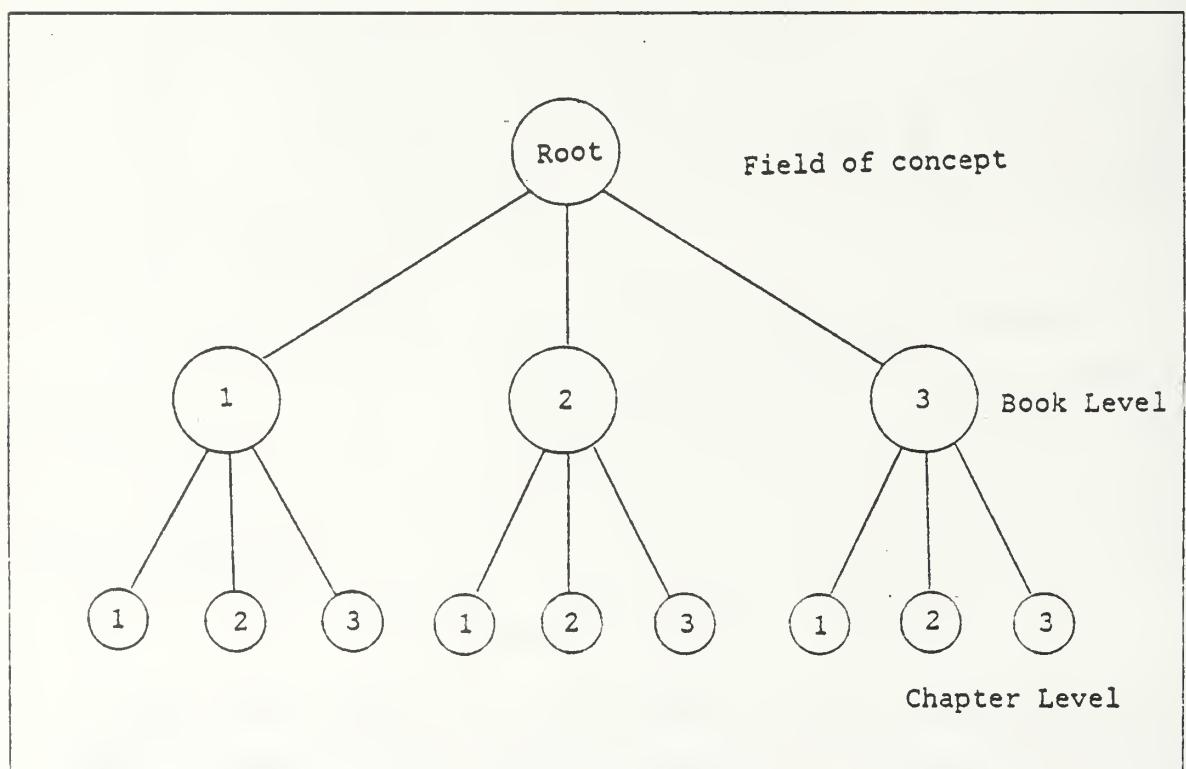
Figure 4.2   The Logical Tree Structure of The Questions Bank

24

The root node represents the domain of knowledge (field of aviation). Figure 4.3 represents the root node.
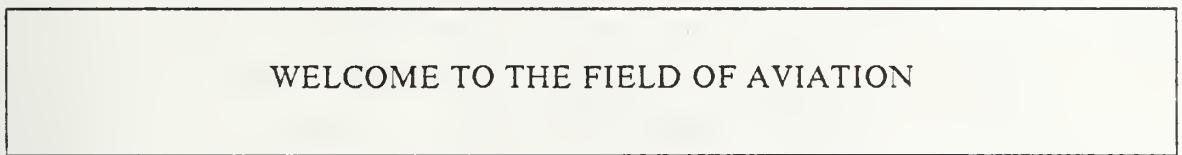
```
┌─────────────────────────────────────────────────────────────┐
│                                                               │
│              WELCOME TO THE FIELD OF AVIATION                 │
│                                                               │
└─────────────────────────────────────────────────────────────┘
```

Figure 4.3   The Representation of Root Node

From the root, if the student enters the command to continue ("press any key to continue"), he/she is going to the next level which is book/concept level. The menu of book/concept is then presented (Figure 4.4).
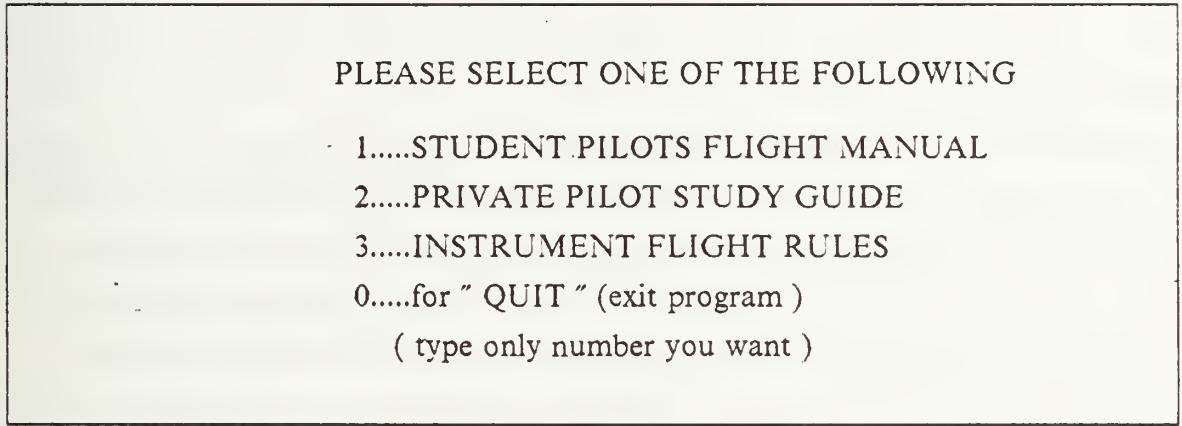
```
┌─────────────────────────────────────────────────────────────┐
│                                                               │
│              PLEASE SELECT ONE OF THE FOLLOWING               │
│                                                               │
│               1.....STUDENT PILOTS FLIGHT MANUAL              │
│               2.....PRIVATE PILOT STUDY GUIDE                 │
│               3.....INSTRUMENT FLIGHT RULES                   │
│               0.....for " QUIT " (exit program )             │
│                  ( type only number you want )               │
│                                                               │
└─────────────────────────────────────────────────────────────┘
```

Figure 4.4   Menu of Valid Concepts in The System

The user/student can choose one of these nodes which is the book/concept number. After the user/student selects a number (node), he/she is going to the next level of chapter, and the menu of chapter is presented (Figure 4.5).

The user/student may again choose one of the nodes which is the chapter number. Each node (number) in this level represents chapter of the selected book which contains test questions. We see that, The number of books is the number of children of the root node (field of aviation). The book/concept is the parent of the chapters and the number of chapters is the number of children of that book, By using tree structure, choosing any chapter to test is like traversing downward the tree from the root. The Testing program handles a domain of knowledge (field of aviation) with 3 concepts and

```
┌─────────────────────────────────────────────────────────┐
│                                                           │
│             " STUDENT PILOT FLIGHT MANUAL "               │
│              SELECT ONE OF THE FOLLOWING                  │
│                                                           │
│                TYPE 1.....for PART - ONE                  │
│                TYPE 2.....for PART - TWO                  │
│                TYPE 3.....for PART - THREE                │
│             TPYE 0.....for SELECTING ANOTHER CONCEPT      │
│                                                           │
└─────────────────────────────────────────────────────────┘
```

Figure 4.5   Menu of Valid Chapters

3 chapters in each concept for an illustration of the logical tree structure. It was decided to use text files to store questions; each file contains questions, correct answers, categories of questions and comments for each test (see Figure 4.6 for explanation).

In order to store the student's performance, an array of records is used, and stored in main memory by defining it as a global variable. One record per chapter/test contains all student's performance about that test. It contains fields for how many questions in the test and, in each category of question, the number of correct and incorrect answered, number of questions completed, the names of concepts and ideas, and also detailed performance in each category of questions (*see record Chap in Appendix E*). The variable *Chapter* (see Appendix E) is of type array. Each element of the array contains one record of student's performance to be used by the program to diagnose the student according to the proposed model and to provide scoring and commentary at the end of the session.

C.   TESTING SYSTEM CONFIGURATION AND DESCRIPTION

The test subject chosen was Pilot Knowledge Evaluation. The program Testing is written in Waterloo Pascal on the IBM mainframe (IBM 3033 AP). The program as presented is designed for a pilot or student pilot. The student/user is assumed to have already learned the basics through traditional classroom lecture techniques or by himself and now needs to test his/her knowledge. Questions about various concepts are stored in the Data Base component so that the student/user can select any concepts which he/she has already learned and be evaluated as to whether or not he/she has

absorbed the instructional material as well as to be diagnosed as to where he/she has difficulties. The user is only required to know the minimal knowledge of how to operate a computer terminal.

As the student is going through the test, his/her performance will be recorded in the Temporary Results component, and (after a certain amount of questions have been asked which are required by the control strategy), the system will determine the probability of his/her ability to answer the rest of the questions. If this probability(P) exceeds the upper cut-off criterion (A = 0.9), he/she is assumed to have mastered the lesson. He/she is exempted from finishing that lesson, and may continue onto the next. If the probability is below the lower cut-off criterion (B = 0.4), he/she is deemed to have insufficient knowledge about the lesson. The control strategy will either suggest the student restart that lesson again or select another concept to study. If P is in between the two cut-off criteria, he/she is assumed to be average, thus, further questioning is required. At the end of the session, when the student wants to exit the program, appropriate conclusions are made about the degree of proficiency of the student, his/her strengths, weakness, and misconceptions, and remedial strategies are suggested. Appendix B and C demonstrate Testing.

D. TESTING SYSTEM DETAILED DESIGN AND DESCRIPTION

1. Design of the File

The question set for each test is designed as a separate text file. When a student selects a test, a specific text file contains only questions for that test is called. As explained early in Chapter 3 (Development of questions for Testing System), the questions are: (1) separated into 3 sets: critical, complementary and supplementary questions; (2) grouped into 3 types: procedural, calculative and physical questions; (3) determined individually by the system as an easy or difficult question. These questions cover seven aspects of "Part1- Before The Flight", in "Student Pilot's Flight Manual" [Ref. 18]. The design of this file is shown in Figure 4.6.

In Figure 4.6, the line under the correct answer contains one number and two characters. The first digit indicates (for the above question) which of the seven areas the question refers to {Part1- Before The Flight}. Codes for these areas are shown in Figure 4.7. The character following the first number identifies the question type, procedural, or calculative, or physical question. Codes for these types are shown in Figure 4.8. The last character on the line under correct answer indicates the above question is considered an easy or difficult one ("e" stands for an easy question, "x"

```
File : File1 Text A1
45        (* number of questions in this file *)
14        (* number of critical questions *)
16        (* number of complementary questions *)
15        (* number of supplementary questions *)
1) ...............(* Body of question *)...........
        ...............................................
                1.....................................
                2.....(* Four multiple choices *) ....
                3.....................................
                4.....................................
S         (* symbol indicates end of question1 *)
2         (* correct answer *)
2se       (* to be explained *)
    ...(* comment & suggestion for question 1 no more than one line *)...
2) ...............................................
        ...............................................
                1.....................................
                2.....................................
                3.....................................
                4.....................................
S
2
3px
    ...(* comment & suggestion for question 2 no more than one line *)...
3).............................
        ...............................
        ...............................
```

Figure 4.6   The Design of File Containing Questions

stands for a difficult question). For example, "2se" on the line under correct answer in question 1 of Figure 4.6 means the above question is in the area of "The Airplane and How It Flies" (2), it is a physical question (s), and considered an easy one (e). An example of this file and questions are shown in Appendix A.

Codes #	Meaning
"1"	" Starting to Fly "
"2"	" The Airplane and How It Flies "
"3"	" Cockpit-Instrument and System "
"4"	" Preflight Check "
"5"	" Starting The Airplane "
"6"	" Taxiing "
"7"	" Pre Take-off and Cockpit Check "

Figure 4.7   Codes for Areas in Part 1- Before The Flight

Codes #	Meaning
"p"	" Procedural question "
"c"	" Calculative question "
"s"	" Physical question "

Figure 4.8   Codes for Types of Questions

2. Description of Operation and Examples

The proposed program was designed using a top-down stepwise refinement methodology. Figure 4.9 shows the steps of operation in Testing.

The operation of Testing begins with a students need to test his/her knowledge of a concept which is provided in the domain of knowledge (field of aviation prepared in the system). The student is provided a listing of concepts, which gives him/her a list of valid ideas provided in the system. Figure 4.4 shows the menu of concepts.

29
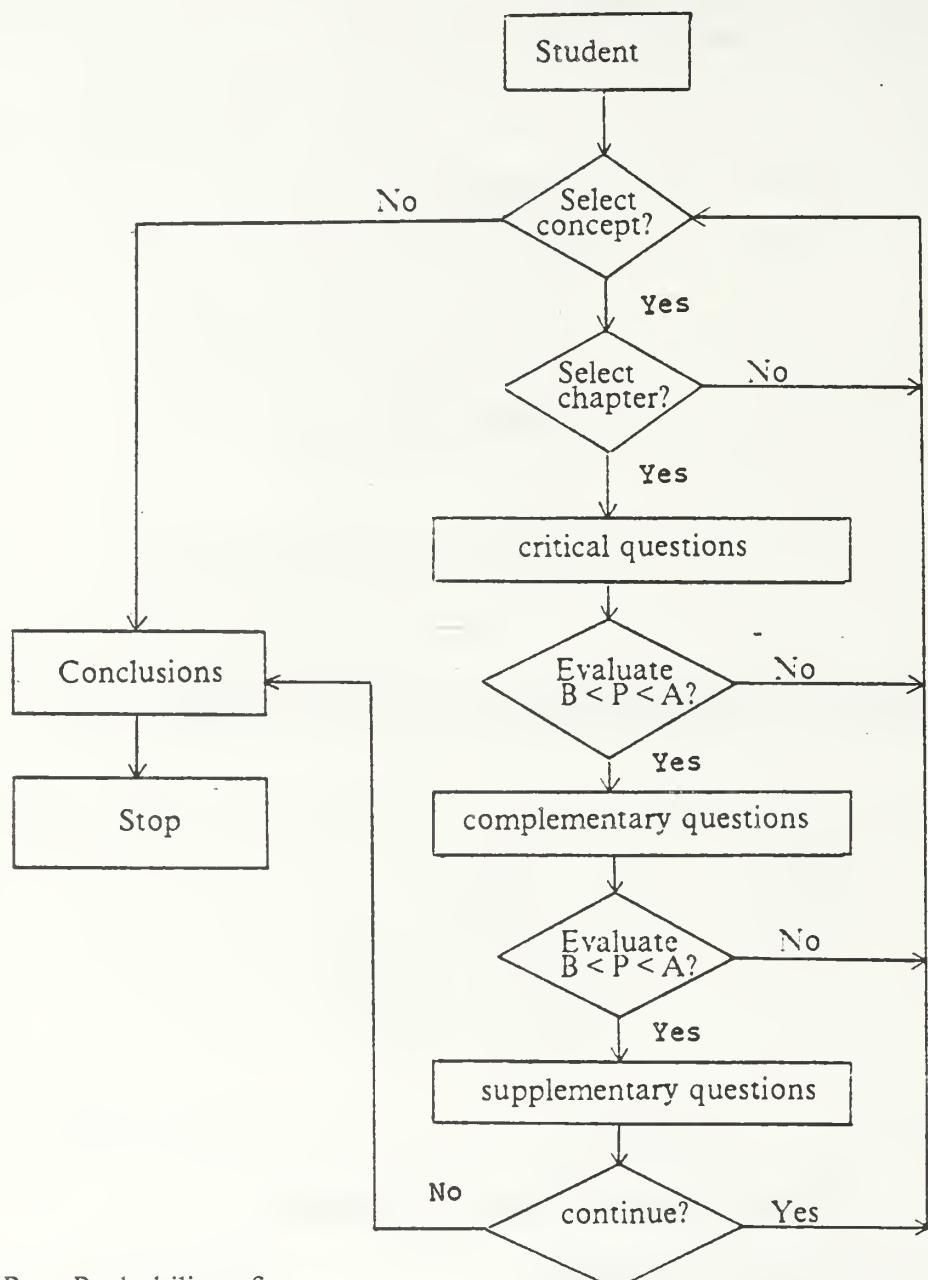
Figure 4.9   Testing System's Flow Chart

Once the student entered the desired concept to test upon, information about the test's processes is presented. This gives the student some idea of what is going to be presented during the session. When the student enters the command to continue ("press any key to continue"), the list of chapters within his/her selected concept is provided so he/she may choose the one that he/she wants to work on. A sample list is shown in Figure 4.5.

The very last step of the system is presentation of the final score and comments which is made after the student enters the quit command ("type.. 0"). This shows the student's performance in the test (mistakes that he/she made and percent grade in every aspect of the selected chapter). If the percent grade in any aspect is less than 50, the system gives notice to the student of his/her mistakes and areas of difficulty. The system also records the student's performance permanently in a historical file for future use (student ID#, name, date, and grade in every test). Previous historical data is provided so that the student will know his/her status to date. Examples of score and commentary are shown in Appendix B and C. Appendix B demonstrates the performance of a bright student, Appendix C demonstrates the performance of a weak student.

V. CONCLUSIONS

A. DISCUSSION OF THE RESEARCH

In this thesis, a modification of the standard CAI system is presented. This modification is unique in that it combines the standard tutorial strategy of CAI and the Bayesian Analysis to tailor a lesson to better suit the student's ability or proficiency in mastering the subject being taught. It demonstrated the feasibility and benefits of using an intelligent CAI system to assist learning strategy in any domain of knowledge. The use of Pascal as a practical language for the intelligent CAI systems was shown. The benefits of Testing are:

- Provides a simple approach.
- Easy to develop and implement.
- Suited to courses where a repetitive practice is required to learn a skill or a concept.
- Allows a student to work through a large number of problems to gain an intuitive understanding of the solutions procedures.
- Uses little core memory, runs fast, and can be used in portable microcomputer.
- Provides self-testing or self-evaluating.
- Reduces the instruction time of bright students.
- Provides the slow ones to progress along at their own pace.
- Accelerates the learning process through identification of weak areas.

B. TESTING SYSTEM LIMITATIONS

Testing is a prototype program that was designed to demonstrate the concept that an intelligent CAI system using the Bayesian Diagnostic model could provide valuable assistance in tutoring the student to learn the lesson. This prototype program demonstrates concepts in the field of aviation in which the author has experience.

Due to time limitations, the Testing program was developed using a limited number of concepts and test questions. One test question set has been made from Chapter 1 of "Student Pilot's Flight Manual" [Ref. 19]. The program was designed to handle 3 books (see Figure 4.4) with 3 tests in each book. With this designed, it was possible to demonstrate the functions of the proposed model. Since this program was written in Waterloo Pascal on the IBM mainframe (IBM 3033 AP), it can be used in

some personal computers, if the program is rewritten so that it can be used on more personal computers. It will be more practical for some small organizations to use this syetem. Another limitation is this program is not graphics-interfaced which would help the user/student in learning. The system could display pictures to describe some ideas that cannot be done well by text.

C. FUTURE PROGRAM IMPROVEMENTS

The purpose of this document is to demonstrate the proposed model in Chapter 3. With the time limitations, the program handles only a few concepts. The data structures used in the program are arrays, records, and files. The program can be easily modified to handle many more concepts by increasing number of arrays to store the records of student performance. It is easy to increase the arrays of records by changing the global constants such as *maxbook, and maxpart* (see Appendix E) that indicate maximum number of concepts and maximum number of tests in a concept respectively. The Testing program is composed of only a few modules or procedures. It is easy for an experienced programmer to understand the functions of each module. It would take only a few days to develop Testing program to handle more concepts. The major tasks are to do some changes in those procedures that determine which concept is selected by the user/student and present him/her with the test questions. The procedure *Booklist*, *Beginsession*, *Whatchapter*, and *Selectchapter* must be modified by adding numbers that identify the concepts in the same fashion as previously done (see Appendix E). The only time-consuming task is to develop the questions for the concepts that are needed. Time to complete the questions is dependent on how many concepts are required and the people who develop the questions.

For better program improvements, a linked-list representation is recommended to store the student's performance instead of arrays. One advantage of a linked-list representation over the arrays used in Testing is that less memory space can be used. For better tutoring, a graphics interface is recommended.

APPENDIX A
ILLUSTRATION OF QUESTIONS USED TO DEMONSTRATE TESTING SYSTEM

The following questions are drawn from Student's Pilot Flight Manual.

```
45
14
16
15
1) Which one of the following statements is correct concerning "torque"
     corrections (U.S.-built engines)?
          1.....In a climb, left rudder is necessary to keep the airplane
                 straight.
          2.....In a climb, right rudder is necessary to keep the airplane
                 straight.
          3.....In a dive, right rudder is necessary to keep the airplane
                 straight.
          4.....When power is added abruptly, the airplane will tend to
                 roll to the right.
$
2
2se
  'Torque' see chapter 2 ,fig. 2-9, SPFM.
2) Which one of the following statements concerning the Four Forces
     is correct?
          1.....In straight and level flight, all Four Forces have
                 equal values.
          2.....In a climb, lift is greater than weight.
          3.....In straight and level unaccelerated flight, thrust is
                 greater than drag.
          4.....Lift acts perpendicular to the relative wind, whether
                 the airplane is climbing, flying, flying straight and
                 level, or gliding.
$
4
2se
  'Four Forces'  See chapter 2, fig. 2-3, SPFM.
3) True altitude is height above the sea level, and absolute altitude
     is height above the surface. With this in mind, choose the correct
     answer to the following: An airplane is flying over the ocean at
     an actual height of 10,000 feet over the water
        (altimeter setting and temperature unknown). It is flying at its
          1.....density and absolute altitudes.
          2.....pressure and absolute altitudes.
          3.....true and absolute altitudes.
          4.....true and density altitudes.
$
3
3px
  'Study all four types of altitude', in chapter 3.
4) You are flying at an indicated airspeed of 120 knots at 6,000 feet
     (assume no instrument error) the outside air temperature is +3
     degrees celsius. Your computer is back on the counter at the airport,
     but using rule of thumb you can estimate that your true airspeed
     is approximately
          1.....140 mph.
          2.....134 knots.
          3.....124 mph.
          4.....144 knots.
$
2
3cx
  see chapter 3, 'Rule of Thumb', page 134, SPSG.
```

5) If you are referring to the magnetic compass on a heading of east,
   an acceleration of airspeed(maintaining a constant altitude)
   will result in
           1.....a more northerly indication.
           2.....a more southerly indication.
           3.....no change in compass indication, since it is correct on
                 east or west heading.
           4.....northerly turning error.
$
1
3sx
 'ANDS', p.45 (SPFM) and p.134,#308 (SPSG).
6) During the pretake-off check after checking the mags, you pull
   the carburetor heat ON at 1800 rpm(or the manufacturer's
   recommended number). There is no change of rpm (plus or minus).
   when you pull it ON or for 30 seconds thereafter. This is a sign :-
           1.....carburetor ice was present before the heat was applied.
           2.....there was no carburetor ice present before the heat was
                 applied.
           3.....the carburetor is not getting heat from the system.
           4.....you did leave the heat on long enough to get an effect.
$
3
4px
 'Function of carburetor', see item G in Fig. 4-M,Fig. 4-2 in SPFM.
7) After starting your airplane's engine on a cold morning, you taxi out
   and make your pretake-off check. After the mag and carburetor heat
   check you reduce the power to idle and find that even though the
   engine is warmed up it lopes and run very rough, nearly quitting but
   not quite. A likely problem is that the
           1.....fuel selector valve has been in the OFF position since
                 before starting.
           2.....cabin heat is ON.
           3.....primer is not in and locked.
           4.....electric fuel pump is not ON.
$
3
5sx
 'Primer system', Fig. 23-17,SPFM.
8) It is a very hot day and the trainer you are going to use has just
   come down from a flight. After thorough preflight check you get
   fastened in. After using your usual procedures you holler CLEAR
   and hit the starter; the propeller turn over and over with no
   indication that the engine wants to start. This is probably because
           1.....the mixture is not rich enough.
           2.....the engine is loaded(flooded).
           3.....the carburetor heat is not ON.
           4.....the engine has cooled too rapidly.
$
2
5se
 'Hot start procedure', in chapter 5, SPFM.
9) You hit the starter of your trainer, and as the propeller turns over
   (the engine hasn't started running well yet), you note that
   spectators are starting wide-eyed at the bottom area of the cowling.
   Your initial best move is to
           1.....open the window and state firmly that it is impolite to
                 stare.
           2.....pump the throttle to help assure that plenty of fuel is
                 getting to the carburetor.
           3.....keep the engine turning over and turn off the mixture
                 and fuel.
           4.....stop the starting procedure until they stop staring.
$
3
5se
 see Pilot Operation Handbook, 'starting procedure'.
10) Hold lines are a combination of two solid and two dashed lines
    crossing the taxiway before it intercepts a runway. The taxi
    guidance line extends through the hold lines and onto the runway.

This means that
    1.....you don't have to slow up before taxiing onto the
            runway at uncontrolled airports.
    2.....you only to recognize hold lines at controlled airports
            because ground control will give or refuse clearance to
            cross them.
    3.....hold lines should be recognized at all airports. You
            will need ATC clearance to cross them at controlled airport
            at uncontrolled airports you should always hold and check
            for traffic before going onto the runway.
    4.....the taxi line has precedence over the hold lines at all
            airports.
$
3
6px
'read chapter 25 of SPSG, Flying the cross country'.
11) You are taxiing a tricycle-gear airplane with a strong wind from
    the right rear. The control positions should be as follows:
    1.....Control wheel forward and turned right, rudder as
            necessary to keep the airplane straight.
    2.....Control wheel forward and turned left, rudder as necessary
            to keep the airplane straight.
    3.....Control wheel aft and turned left, full left rudder.
    4.....Control wheel forward and turned left, full right rudder.
$
2
6sx
'study technique of taxiing in a strong wind.'
12) When taxi downwind with strong wind, you should:
    1.....hold the wheel forward so that the wind strike the down
            elevator and keeps the tail from rising.
    2.....hold the wheel back so that the wind force the propwash
            hold the tail down.
    3.....taxi with engine idle, use brake to keep speed down
            because of the wind forces the airplane to be fast.
    4.....taxi with small power and use brake to keep speed.
$
1
6sx
'How to taxi downwind with strong wind', see SPFM, p.37
13) You are checking the magnetos at the run-up spot and inadvertently
    turn the switch to OFF. The best move for you before turning it
    back on is to
    1.....pull the carburetor heat ON.
    2.....close the throttle.
    3.....pull the mixture to idle cut-off.
    4.....switch tanks.
$
2
7px
'Accidentally turning magnetos switch of during run-up'.
14) You have just completed the run-up. Just before starting to taxi
    into the take-off position, you realize that the fuel gage shows
    the tank you ran up on is very low in fuel while the other is
    nearly three quarters full. You recall this was that the case when
    you visually checked the tanks, but you forgot to put the selector
    on the fullest tanks. You should
    1.....switch tanks and continue the take-off.
    2.....keep the airplane on the tank you did the run-up on and
            go ahead and take-off.
    3.....switch tanks and run the engine up to at lease 1500 rpm
            for a minute or more, or redo the mag and carburetor heat
            check.
    4.....do none of the above.
$
3
7px
'Very low fuel gage before taking-off'.
15) You are solo and just lifting off (with the runway gone behind)
    it sounds as if the engine is banging itself to destruction. The

airplane is performing as usual (rpm,pressures, and temperatures
are okay). The best move in this case is to
       1.....maintain control of the airplane as you continue to climb.
       2.....land straight ahead immediately.
       3.....turn back to the airport without delay.
       4......start dumping ballast overboard.
$
1
7px
 'Rough engine right after taking-off'.
16) Which of the following combinations would most likely cause loss of
    power and/or a rough-running engine during the pretake-off check?
       1.....Airport at a high elevation, high outside air temperature,
             mixture full rich, and carburetor heat ON.
       2.....Airport at sea level, low outside air temperature,
             mixture full rich, and carburetor heat ON.
       3.....Airport at a high elevation, high outside air temperature,
             mixture leaned, and carburetor heat OFF.
       4.....Airport at sea level, high outside air temperature,
             mixture full rich, and carburetor heat OFF.
$
1
7sx
 'causes of losing power and/or rough engine in pre-takeoff'
17)You are a private pilot and plan to fly a trainer into a 2000-foot
    farm strip. You check the Pilot's Operating Hand-book and see that
    the airplane can land or take-off with about 500 feet to spare under
    the existing conditions. At the end of the pre-takeoff check you
    close the throttle and the engine idles at 800 instead of the
    recommended 600 rpm. You should
       1.....continue the flight.
       2.....taxi back and get the mechanic to reset the idle to the
             recommended value before flying.
       3.....continue the flight but expect a steeper approach and short
             landing roll because the extra windmilling effects will
             create drag.
       4.....be sure to notify an instructor or mechanic about the high
             idle value after you get back to the home airport.
$
2
7px
 'idle rpm exceeded at the end of pre-takeoff check'.
18) Taking off at higher altitudes and temperatures for a particular
    airplane (at a given weight) will require
       1.....more runway because the air is less dense.
       2.....less runway because the air is less dense.
       3.....the same amount of runway because the less dense air
             create less drag, making up for any power loss.
       4.....more runway because the air becomes more dense with
             increase in altitude. What horse power is required to raise
             55 pounds to a height of 6 feet in 3 seconds?
$
1
2sx
 'taking off at high altitude and temperature, p.6 SPFM'.
19) What horse power is required to raise 55 pounds to a height of
    6 feet in 3 seconds?
       1.....1/10.
       2.....1/50
       3.....1/5.
       4.....1.5.
$
3
2ce
 'how to determine horse power at a situation'.
20) The drag resulting from lift being produced is
       1.....parasite drag.
       2.....interference.
       3.....induced.
       4.....gyroscopic.

$
3
2sx
'DRAG', and see Fig. 2-14, 2-16 in SPFM.
21) Which of the following statements is correct concerning drag?
        1.....Induced drag increases as the airspeed increases.
        2.....If parasite drag is 200 pounds at a straight adn level
                speed of 100 knots, at 200 knots it will be 800 pounds.
        3.....Skin friction drag increases with increased airspeed.
        4.....Induced drag decreases as the airspeed decreases.
$
2
2cx
'study the definition of DRAGS in SPFM.'
22) You are flying on a solo cross-country in a trainer and realize
    that the oil pressure gage is at ZERO. Your best initial move
    would be to
        1.....land immediately;the engine will stop within 3 minutes.
        2.....watch the oil temperature gage for an increase in
                temperature as you fly toward an airport or better terrain.
        3.....pay no attention to it but continue your flight.
        4.....land immediately to 7700 on the transponder
                (if you have one) and call MAYDAY on 121.5.
$
2
3px
'emergency procedure for zero oil pressure in flight.'
23) If the engine-driven vacuum pump fails in the average trainer,
    you would expect to lose the use of
        1.....airspeed and altitude indicators.
        2.....altimeter and altitude indicators.
        3.....altitude indicator and heading(direction)indicator.
        4.....vertical speed indicator and turn coordinator.
$
3
3sx
'loss of engine-driven vacuum pump, see Fig. 3-22, SPFM.'
24) At a calibrated airspeed of 150 knots at a density altitude of
    10,000 feet, you want to set up a standard rate turn. Using the
    altitude indicator for bank angle, you would set up a bank of
    approximately
        1.....15 degrees
        2.....25 degrees
        3.....35 degrees
        4.....10 degrees
$
2
3cx
'standard rate turn at any particular airspeeds.'
25) Which one of the following items would require the most
    electrical current when in operation?
        1.....Landing light.
        2.....Turn and slip indicator.
        3.....Flashing beacon.
        4.....Navigation receiver.
$
1
3se
'electrical system.'
26) Which of the following would most likely result in a sudden increase
    in oil consumption in the engine of an airplane?
        (There are no outside leaks in evidence.)
        1.....Eroded spark plug electrodes.
        2.....Worn or broken piston rings.
        3.....A poor adjusted carburetor.
        4.....A coagulated frammis.
$
2
4sx
'sudden increase in oil consumption.'

27) As an all-around procedure you should
        1.....deliberately run a tank dry to use all the fuel only if it
                is a fuel injection system.
        2.....expect a delay of 1-2 seconds in power after the selector
                is switched from an empty to a usable tank with the fuel
                injection system.
        3.....deliberately run a tank dry only at altitude above 1500
                feet MSL
        4.....never deliberately run a tank dry.
$
4
4px
 'deliberately run a tank dry during flight?'
28) You have your private certificate and are taking a friend for a
        first ride. You find that the battery is dead. It's late and
        there is no one else at the airport, but jumper cables are
        available. Also, you have checked out to hand-crank the airplane.
        Your next move would be to
        1.....carefully show the friend how to hold the brakes and turn
                the ignition switch ON while you hand-crank the airplane.
        2.....show the friend how to hold the brakes and use the starter
                while you set up jumper cables from your car.
        3.....Stay in the airplane to start it, and the friend can work
                the jumper cables (revving up the car and then removing
                the cables after the start).
        4.....cancel the flight until the airplane's electrical system
                is back to normal.
$
4
5px
 'procedure for dead battery before taking off.'
29) A soft or spongy brake is usually the result of having air in the
        system. To get braking action you should do which of the following?
        1.....Pump the brake pedals.
        2.....Hold the pressure but do not pump.
        3.....Pull the parking brake handle outward.
        4.....Stay completely off the brake.
$
1
6px
 'procedure for having a soft or spongy brake while taxiing.'
30) The "best" fuel-to-air ratio is approximately
        1.....1 to 15.
        2.....15 to 1.
        3.....1 to 7.
        4.....20 to 1.
$
1
5px
 'fuel-to-air ratio.'
31) The angle between the relative wind and the chord line of the
        airfoil is the
        1.....angle of incidence.
        2.....angle of attack.
        3.....planning angle.
        4.....angle of climb or glide.
$
2
2pe
 'angle of attack', see Fig. 2-5, SPFM.
32) Brake horsepower is
        1.....Horsepower being developed at the crankshaft.
        2.....Horsepower developed by the propeller in moving the
                airplane through the air.
        3.....measured as pounds of thrust horsepower in older airplane.
        4.....always the same as thrust horsepower in newer airplane.
$
1
2pe
 'brake horse power', p.6, SPFM.

33) Gravity always acts
        1.....opposite to weight.
        2.....parallel but opposite to lift.
        3.....perpendicular to the total drag.
        4.....toward the center of the earth.
$
4
2pe
 'gravity, see "Weight", SPFM.'
34) In computing airplane performance, the pilot or engineer uses
        1.....density altitude.
        2.....true altitude.
        3.....absolute altitude.
        4.....indicated altitude.
$
1
3ce
 'computing airplane performance.'
35) Indicated airspeed corrected for instrument and position error is
        1.....true airspeed.
        2.....calibrated airspeed.
        3.....equivalent airspeed.
        4.....static pressure.
$
2
3ce
 'understanding all four kinds of airspeed.'
36) Equivalent airspeed corrected for density altitude effects is
        1.....calibrated airspeed.
        2.....indicated airspeed.
        3.....true airspeed.
        4.....density airspeed.
$
3
3ce
 'the relations among the airspeeds.'
37) The job of pitot tube is to admit what pressure or pressures to the
    airspeed indicator?
        1.....Dynamic only.
        2.....Static only.
        3.....Static and vertical changes.
        4.....Dynamic and static.
$
4
3pe
 'the function of Pitot tube,' see POH.
38) A job of the alternater is to
        1.....provide a spark to each magneto as needed.
        2.....provide vacuum for the gyro instruments.
        3.....run the electrical components and keep the battery charged.
        4.....alternate the power from the brake system to the electrical
                system.
$
3
3pe
 'the functions of alternater, check POH.'
39) The majority of turn and slip indicators or coordinators in current
    smaller airplanes are powered by the
        1.....static pressure system.
        2.....engine-driven vacuum pump.
        3.....hydraulic system.
        4.....electrical system.
$
4
3pe
 'soures power of instrument indicators.'
40) The vertical speed indicator operates on the principle of
        1.....change in outside atmospheric pressure.
        2.....absolute atmospheric pressure.
        3.....precession.

```
                    4.....dynamic pressure from the pitot tube.
$
1
3pe
 'how the vertical speed works.'
41) The addition of carburetor heat
            1.....leans the mixture.
            2.....has no effect on the mixture.
            3.....riches the mixture.
            4.....cause oil pressure fluctuations.
$
3
4se
 'understand the uses of carburetor heat.'
42) An advantage in the fuel injection system over the carburetor is
     that it
            1.....has better fuel distribution to the cylinders.
            2.....does not require an alternate air system.
            3.....can be used best with fixed-pitch propellers.
            4.....has a simpler oil system.
$
1
4pe
 'the advantages of fuel injection system.'
43) Smoking in an airplane
            1.....is useful in finding out the wind direction by blowing
                    the smoke out of the window.
            2.....may cause you to be thrown overboard by a nonsmoker.
            3.....can cause clogging of the vacuum system air filter.
            4.....can clog the carburetor heat.
$
3
4se
 'causes of carburetor clogged during flight.'
44) You start the engine and check the oil pressure gage. The needle
     doesn't move right away, but you figure that since it's summer
     you can expect it to come to normal operation within
            1.....42 hours and 26 minutes.
            2.....5 seconds.
            3.....30 minutes.
            4.....30 seconds.
$
4
5pe
 'operating of oil pressure indicator after starting the engine.'
45) Taxiway marking are
            1.....white dashed lines.
            2.....yellow solid lines.
            3.....yellow dashed lines.
            4.....white solid lines.
$
2
6se
 'Taxi Marking', see SPSG P.19
¢
```

APPENDIX B
DEMONSTRATION OF BRIGHT-STUDENT PERFORMANCE

```
************************************************
*                                              *
*                                              *
*        WELCOME   TO   THE FIELD   OF   AVIATION   *
*                                              *
*                                              *
************************************************
please type your number
your name ?
today date,(ex. 10 Apr 87)
"press enter to continue"

*****************************************************
*                                                   *
*         PLEASE SELECT ONE OF THE FOLLOWING         *
*                                                   *
*      1.....STUDENT PILOTS FLIGHT MANUAL            *
*                                                   *
*      2.....PRIVATE PILOT STUDY GUIDE               *
*                                                   *
*      3.....INSTRUMENT FLIGHT RULES                 *
*                                                   *
*      0.....for  " QUIT "  (exit program)           *
*                                                   *
*         (Type only number you want )               *
*                                                   *
*****************************************************
:1

*****************************************************
*                                                   *
*      In each test which you will work on :-        *
*                                                   *
*    1)   There are 3 sets of questions.             *
*    2)   Each set is different in level of difficulty *
*    3)   The most difficult set will be presented   *
*         first and then the next two sets.          *
*    4)   You will be evaluated after completing each *
*         of the first two set of questions.         *
*    5)   If you are about average, all questions    *
*         will be presented, otherwise you do not    *
*         have to do all of the questions.           *
*                                                   *
*****************************************************

   " press any key to continue "

*****************************************************
*                                                   *
*              THE FOLLOWING ARE QUESTIONS           *
*                                                   *
*                      FROM                          *
*                                                   *
*          " STUDENT PILOT FLIGHT MANUAL "           *
*                                                   *
```

```
        *                                                    *
        ******************************************************

    " press any key to continue "


        ******************************************************
        *                                                    *
        *          " STUDENT PILOT FLIGHT MANUAL "           *
        *            SELECT ONE OF THE FOLLOWING             *
        *                                                    *
        *        TYPE  1..... for   PART - ONE               *
        *        TYPE  2..... for   PART - TWO               *
        *        TYPE  3.....for   PART - THREE              *
        *     TYPE  0..... for   SELECTING ANOTHER CONCEPT   *
        *                                                    *
        ******************************************************
  : 1



              ******************************************
              *                                        *
              *   THIS IS THE BEGINNING OF THE LESSON   *
              *                                        *
              ******************************************

    " press any key to continue "
```

1) Which one of the following statements is correct concerning "torque"
   corrections (U.S.-built engines)?
           1.....In a climb, left rudder is necessary to keep the airplane
                  straight.
           2.....In a climb, right rudder is necessary to keep the airplane
                  straight.
           3.....In a dive, right rudder is necessary to keep the airplane
                  straight.
           4.....When power is added abruptly, the airplane will tend to
                  roll to the right.

    Your answer is   1

    ```
    * You are wrong, answer is    2 *
    ```
    _____

2) Which one of the following statements concerning the Four Forces
   is correct?
           1.....In straight and level flight, all Four Forces have
                  equal values.
           2.....In a climb, lift is greater than weight.
           3.....In straight and level unaccelerated flight, thrust is
                  greater than drag.
           4.....Lift acts perpendicular to the relative wind, whether
                  the airplane is climbing, flying, flying straight and
                  level, or gliding.

    Your answer is   4

    ```
    *  You are correct      *
    ```
    _____

(* removed question 3 through question 13 *)

14) You have just completed the run-up. Just before starting to taxi
into the take-off position, you realize that the fuel gage shows
the tank you ran up on is very low in fuel while the other is
nearly three quarters full. You recall this was that the case when
you visually checked the tanks, but you forgot to put the selector
on the fullest tanks. You should
    1.....switch tanks and continue the take-off.
    2.....keep the airplane on the tank you did the run-up on and
          go ahead and take-off.
    3.....switch tanks and run the engine up to at lease 1500 rpm
          for a minute or more, or redo the mag and carburetor heat
          check.
    4.....do none of the above.

Your answer is   3
─────────────────────────
* You are correct   *
─────────────────────────

    number question in critical set =   14
    number correct answers   =   13
    number incorrect answers   =   1
The probability of your ability is =   0.995

───
        You have not yet completed all the questions.
    You will now be evaluated on what you have done so far.
            THERE ARE 45 QUESTIONS.
            14 critical questions.
            16 complementary questions.
            15 supplementary questions.
        you have completed only the critical questions set.
            correct answers   =   13
            incorrect answers   =   1
            " YOU ARE ABOVE AVERAGE, EXCELLENT! "
* * Be careful on easy questions! **
You have done 1 mistakes about physical question.
───

            ****
            *   *
            *   YOUR KNOWLEDGE IN THE PREVIOUS LESSON   *
            *   IS SATISFACTORY   *
            *   YOU ARE FINISHED WITH THIS LESSON   *
            *   *
            ****

    " press any key to continue "

            ****
            *   *
            * THIS IS THE END OF THE LESSON *

44

```
                    *                                    *
          ***********************************
```

" press any key to continue "

```
          ************************************************
          *                                              *
          *        DO YOU WANT TO SEE THE RESULTS         *
          *                                              *
          *         YES    PRESS " ANY KEY "             *
          *          NO     PRESS " N "                  *
          *                                              *
          ************************************************
```
:n

```
          ************************************************
          *                                              *
          *            DO YOU WANT TO CONTINUE            *
          *             ON THE SELECTED CONCEPT?          *
          *                                              *
          *      PRESS " ANY KEY "  to continue          *
          *                      on the selected concept *
          *                                              *
          *      PRESS " N "  to select another lesson   *
          *                      or go to exit           *
          *                                              *
          ************************************************
```
:n

```
          ************************************************
          *                                              *
          *        PLEASE SELECT ONE OF THE FOLLOWING     *
          *                                              *
          *      1.....STUDENT PILOTS FLIGHT MANUAL       *
          *                                              *
          *      2.....PRIVATE PILOT STUDY GUIDE          *
          *                                              *
          *      3.....INSTRUMENT FLIGHT RULES            *
          *                                              *
          *      0.....for  " QUIT "  (exit program)      *
          *                                              *
          *       (Type only number you want )           *
          *                                              *
          ************************************************
```
:0

```
          *****************************************
          *                                       *
          *            FINAL SECTION:             *
          *         SCORES AND COMMENTARY          *
          *                                       *
          *****************************************
```

" press any key to continue "

```
========================================================
                 SCORES AND COMMENTARY
========================================================
             " STUDENT PILOT FLIGHT MANUAL "
             "PART - ONE" (Before Flight)
          THERE ARE 45  QUESTIONS.
              14  critical questions.
              16  complementary questions.
              15  supplementary questions.
       you have answered only the critical questions set.
              Total correct answers  =   13
              Total incorrect answers  =    1
             " YOU ARE ABOVE AVERAGE, EXCELLENT! "
       Your last test was on     10 MAY 1987.
       And the grade was    51.110.
     Your grade today was   92.86.
     Good, you are improving!
              Following is a report of your performance
The Airplane and How It Flies
       Pts = 1,    Total = 2,  Percent Grade =   50.00
Cockpit: Instruments and Systems
       Pts = 3,    Total = 3,  Percent Grade =  100.00
Preflight Check
       Pts = 1,    Total = 1,  Percent Grade =  100.00
Starting The Airplane
       Pts = 3,    Total = 3,  Percent Grade =  100.00
Taxiing
       Pts = 3,    Total = 3,  Percent Grade =  100.00
Pretake-off or Cockpit Check
       Pts = 2,    Total = 2,  Percent Grade =  100.00

 ─────────────────────────────────────────────────────
 * YOUR PROBLEMS AND SUGGESTIONS FOR IMPROVEMENT   *
 ─────────────────────────────────────────────────────

 Question about Procedures:
     Pts = 5,    Total = 5,  Percent Grade =  100.00
 Question about Calculation:
     Pts = 1,    Total = 1,  Percent Grade =  100.00
 Physicals questions:
     Pts = 7,    Total = 8,  Percent Grade =   87.50
     You should review as follows
     1)'Torque' see chapter 2 ,fig. 2-9, SPFM.
    **  Be careful on easy question! **
     You missed  1 of them in the total of  4.
 ─────────────────────────────────────────────────────




              **********************************
              *                                *
              *      HOPE TO SEE YOU AGAIN      *
              *                                *
              **********************************


...execution ends
```

APPENDIX C
DEMONSTRATION OF WEAK-STUDENT PERFORMANCE

```
****************************************************
*                                                  *
*                                                  *
*         WELCOME   TO  THE FIELD  OF  AVIATION     *
*                                                  *
*                                                  *
****************************************************
please type your number
your name ?
today date,(ex. 10 Apr 87)
"press enter to continue"

****************************************************
*                                                  *
*           PLEASE SELECT ONE OF THE FOLLOWING      *
*                                                  *
*         1.....STUDENT PILOTS FLIGHT MANUAL        *
*                                                  *
*         2.....PRIVATE PILOT STUDY GUIDE           *
*                                                  *
*         3.....INSTRUMENT FLIGHT RULES             *
*                                                  *
*         0.....for  " QUIT "  (exit program)       *
*                                                  *
*            (Type only number you want )           *
*                                                  *
****************************************************
:1

****************************************************
*                                                  *
*        In each test which you will work on :-     *
*                                                  *
*     1)  There are 3 sets of questions.            *
*     2)  Each set is different in level of difficulty *
*     3)  The most difficult set will be presented  *
*         first and then the next two sets.         *
*     4)  You will be evaluated after completing each *
*         of the first two set of questions.        *
*     5)  If you are about average, all questions   *
*         will be presented, otherwise you do not   *
*         have to do all of the questions.          *
*                                                  *
****************************************************

   " press any key to continue "

****************************************************
*                                                  *
*           THE FOLLOWING ARE QUESTIONS             *
*                                                  *
*                    FROM                           *
*                                                  *
```

```
        *          " STUDENT PILOT FLIGHT MANUAL "          *
        *                                                    *
        *                                                    *
        ******************************************************
```

" press any key to continue "

```
        ******************************************************
        *                                                    *
        *          " STUDENT PILOT FLIGHT MANUAL "           *
        *            SELECT ONE OF THE FOLLOWING             *
        *                                                    *
        *          TYPE  1..... for  PART - ONE              *
        *          TYPE  2..... for  PART - TWO              *
        *          TYPE  3......for  PART - THREE            *
        *        TYPE  0..... for  SELECTING ANOTHER CONCEPT *
        *                                                    *
        ******************************************************
: 1
```

```
            ******************************************
            *                                        *
            *   THIS IS THE BEGINNING OF THE LESSON   *
            *                                        *
            ******************************************
```

1) Which one of the following statements is correct concerning "torque"
   corrections (U.S.-built engines)?
            1.....In a climb, left rudder is necessary to keep the airplane
                  straight.
            2.....In a climb, right rudder is necessary to keep the airplane
                  straight.
            3.....In a dive, right rudder is necessary to keep the airplane
                  straight.
            4.....When power is added abruptly, the airplane will tend to
                  roll to the right.

     Your answer is   1

     ----
     * You are wrong, answer is   2 *

            (* removed question 2 through question 13 *)

14) You have just completed the run-up. Just before starting to taxi
    into the take-off position, you realize that the fuel gage shows
    the tank you ran up on is very low in fuel while the other is
    nearly three quarters full. You recall this was that the case when
    you visually checked the tanks, but you forgot to put the selector
    on the fullest tanks. You should
            1.....switch tanks and continue the take-off.
            2.....keep the airplane on the tank you did the run-up on and
                  go ahead and take-off.
            3.....switch tanks and run the engine up to at lease 1500 rpm
                  for a minute or more, or redo the mag and carburetor heat
                  check.
            4.....do none of the above.

                              48

```
     Your answer is    3
     ───────────────────────────────
     *   You are correct        *

     ───────────────────────────────



     number question in critical set  =    14
     number correct answers     =     5
     number incorrect answers       =     9
The probability of your ability is =     0.178

───────────────────────────────────────────────────
        You have not yet completed all the questions.
    You will now be evaluated on what you have done so far.
            THERE ARE 45  QUESTIONS.
            14   critical questions.
            16   complementary questions.
            15   supplementary questions.
        you have completed only the critical questions set.
             correct answers    =     5
             incorrect answers   =     9
        " YOU ARE BELOW AVERAGE IN THIS PART "
 * * Be careful on easy questions! **
You have made 3 mistakes about physical question.

───────────────────────────────────────────────────



         ********************************************
         *                                          *
         *      YOU ARE WEAK FOR THIS LESSON        *
         *                                          *
         * STUDY"PART - ONE" (Before Flight)  AGAIN *
         *                                          *
         *     YOU ARE FINISHED WITH THIS LESSON    *
         *                                          *
         ********************************************


  " press any key to continue "



         **********************************
         *                                *
         *  THIS IS THE END OF THE LESSON *
         *                                *
         **********************************




         *************************************************
         *                                               *
         *       DO YOU WANT TO SEE THE RESULTS           *
         *                                               *
         *         YES     PRESS " ANY KEY "             *
         *         NO      PRESS " N "                   *
         *                                               *
         *************************************************
```

```
****************************************************
*                                                  *
*            DO YOU WANT TO CONTINUE               *
*             ON THE SELECTED CONCEPT?             *
*                                                  *
*        PRESS " ANY KEY "  to continue            *
*                        on the selected concept   *
*                                                  *
*        PRESS " N "  to select another lesson     *
*                        or go to exit             *
*                                                  *
****************************************************
```

```
****************************************************
*                                                  *
*         PLEASE SELECT ONE OF THE FOLLOWING        *
*                                                  *
*      1.....STUDENT PILOTS FLIGHT MANUAL          *
*                                                  *
*      2.....PRIVATE PILOT STUDY GUIDE             *
*                                                  *
*      3.....INSTRUMENT FLIGHT RULES               *
*                                                  *
*      0.....for  " QUIT "  (exit program)         *
*                                                  *
*      (Type only number you want )                *
*                                                  *
****************************************************
```

```
***************************************
*            FINAL SECTION:            *
*                                      *
*         SCORES AND COMMENTARY        *
*                                      *
***************************************
```

```
========================================================
                    SCORES AND COMMENTS
========================================================
```

       " STUDENT PILOT FLIGHT MANUAL "

        "PART - ONE" (Before Flight)

    THERE ARE 45 QUESTIONS.
        14   critical questions.
        16   complementary questions.
        15   supplementary questions.

     you have answered only the critical questions set
            Total correct answers =   5
            Total incorrect answers =   9

      " YOU ARE BELOW AVERAGE IN THIS PART "

    Your last test was on   15 MAY 1987.
    And the grade was   92.860.

```
Your grade today was   35.71.
You are weaker than last time,"Study harder"!
          Following is a report of your performance
The Airplane and How It Flies
     Pts =  0,   Total =  2,  Percent Grade =    0.00
Cockpit: Instruments and Systems
     Pts =  1,   Total =  3,  Percent Grade =   33.33
Preflight Check
     Pts =  1,   Total =  1,  Percent Grade =  100.00
Starting The Airplane
     Pts =  1,   Total =  3,  Percent Grade =   33.33
Taxiing
     Pts =  1,   Total =  3,  Percent Grade =   33.33
Pretake-off or Cockpit Check
     Pts =  1,   Total =  2,  Percent Grade =   50.00


*  YOUR PROBLEMS AND SUGGESTIONS FOR IMPROVEMENT *
_____

Question about Procedures
     Pts =  3,   Total =  5,  Percent Grade =   60.00
     You should review as follows
     1)'read chapter 25 of SPSG, Flying the cross country'.
     2)'Accidentally turning magnetos switch of during run-up'.
_____

Question about Calculation
     Pts =  0,   Total =  1,  Percent Grade =    0.00
     You should review as follows
     1)see chapter 3, 'Rule of Thumb', page 134, SPSG.

     ** Study more! You are weak here. **
_____

Physical Questions
     Pts =  2,   Total =  8,  Percent Grade =   25.00
     You should review as follows
     1)'Torque' see chapter 2 ,fig. 2-9, SPFM.
     2)'Four Forces'  See chapter 2, fig. 2-3, SPFM.
     3)'ANDS', p.45 (SPFM) and p.134,#308 (SPSG).
     4)'Primer system',  Fig. 23-17,SPFM.
     5)'Hot start procedure', in chapter 5, SPFM.
     6)'study technique of taxiing in a strong wind.'

     ** Study more! You are weak here. **
    **  Be careful on easy question! **
     You missed  3 of them in the total of  4.
_____




          ***********************************
          *                                 *
          *      HOPE TO SEE YOU AGAIN       *
          *                                 *
          ***********************************


...execution ends
```

```
program Testing(input,output);
(*$$ 200000 *)
(**********************************************************)
(* TITLE :      PILOT_TESTING                           *)
(* AUTHOR :     CDR Yodchai Rugsumruad                  *)
(* Date Written : 1 Apr 87 - 20 May 87                  *)
(*                                                      *)
(* Version      : One                                   *)
(* System Used  : Waterloo Pascal on NPS mainframe      *)
(* I/O Sources  : Data input from secondary storage     *)
(*                and user input from terminal keyboard *)
(* Description  : This program demonstrates an intelligent *)
(*                CAI System for testing, evaluating, and  *)
(*                recommendation the pilot.              *)
(**********************************************************)
                 (*    GLOBAL CONSTANTS    *)
const
     maxchar = 72;           (* for read 1 line of char *)
     maxcomment = 60;        (* for store comments in array *)
     maxchap =  10;          (* Number of chapter to test *)
     uninformed = 0.5;       (* Uninformed Opinion *)
     expert  = 0.7;          (* Expert Opinion *)
     lowweight = 1.0;        (* low level of confidence *)
     highweight = 4.0;       (* high level of confidence *)
     alpha =  0.9;           (* Max cut-off Criteria *)
     beta  = 0.4;            (* Low cut-off Criteria *)
     maxbook = 10;           (* Max number of book to test *)
     maxpart = 10;        (* Max number of chapter or part in each book *)
     maxarea = 10;        (* Max concepts in each chapter or part *)
     maxcat = 3;          (* Max categories or types of questions *)
     maxproblem = 25;(* Max comments in each type of questions *)
type
     ql = packed array(.1..maxchar.) of char;
     chap  = record
          num : integer;
          count,
          correct,
          wrong,
          block,
          crit,
          rcrit,
          wcrit,
          comp,
          rcomp,
          wcomp,
          sup,
          rsup,
          wsup : integer;
        end; (* chap *)
     chapquestion = packed array(.1..maxchap.) of chap;
     board = record
          flag1,
          flag2,
          flag3 : boolean;
          ex,
          av,
          poor  : ql;
        end; (* board *)
     blackboard   = packed array(.1..maxchap.) of board;
     suggest      = record
```

```pascal
                number : integer;
                trob : ql;
                pages : ql;
            end;
    chapcomment  = packed array(.1..maxchap,1..maxcomment.) of suggest;
    topic      = record
            name : ql;
            fname : boolean;
        end;
booktype = packed array(.1..maxbook.) of topic;
subpart    = record
            subname : ql;
            fsub : boolean;
        end;
parttype     = packed array(.1..maxpart.) of subpart;
areainfo = record
            name : ql;
            count,
            wrong,
            correct : integer;
            flag : boolean;
        end;
    areatype = packed array(.1..maxpart,1..maxarea.) of areainfo;
    blockarray    = packed array(.1..maxchap.) of integer;
    countcomment = packed array(.1..maxchap.) of integer;
    commarray = packed array(.1..maxproblem.) of ql;
    problems   = record
            name : ql;
            flag : boolean;
            count,
            correct,
            wrong,
            easy,
            wasy   : integer;
            comm : commarray;
        end;
    problemtype = packed array(.1..maxcat.) of problems;
    nametype = packed array(.1...20.) of char;
    datetype = packed array(.1...15.) of char;
                        (*   GLOBAL VARIABLES   *)

var
        block : blockarray;
        next : boolean;
        rec : blackboard;
        continue : boolean;       (* for beginning or stop of session *)
        comm,
        question : ql;            (* for reading a line of question *)
        correctans,               (* for reading correct answer from file *)
        answer,                   (* for reading answer from screen *)
        ans,                   (* answer y/n from screen to begin session *)
        symbol : char;            (* symbol in file *)
        choose : boolean;
        a,numq,right,wrongq : integer;
        file1, file2, file3, file4 : text;  (* text files of questions *)
        chapter : chapquestion;
        comment : chapcomment;
        numcomment : countcomment;
        book : char;
        b : integer;  (* indicate number of concept *)
        books : booktype;
        part  : parttype; (* record any part in a book *)
        area : areatype; (* record every concept in a test *)
        areas : integer;
        x : real;     (* for percent grade *)
        problem : problemtype;
        p,m,n : integer;
        cats : char;        (* char represents question category *)
        cat : integer;
        pilotnum: integer;
```

```pascal
        pilotname : nametype;
        pilotdate : datetype;
        pilotgrade : real;
        testnum : integer;
        name : nametype;
        todaydate : datetype;
        pilotf : text;
        tempfile : text;
(****************************************************************
 *                   Procedure Initializevalue                *
 * This procedure initializes value of global variables in the program *
 ****************************************************************)
procedure initializevalue;
var
     i,j : integer;
begin
     ans := ' ';
     book := ' ';
     next := true;
     for i := 1 to maxpart do begin
         chapter(.i.).num := 0;
         numcomment(.i.) := 0;
         chapter(.i.).num := 0;
         chapter(.i.).count := 0;
         chapter(.i.).correct := 0;
         chapter(.i.).wrong := 0;
         rec(.i.).flag1 := false;
         rec(.i.).flag2 := false;
         rec(.i.).flag3 := false;
         rec(.i.).ex := '   ';
         rec(.i.).poor   := '   ';
         chapter(.i.).block := 0;
         books(.i.).name := '   ';
         books(.i.).fname := false;
         part(.i.).subname := '   ';
         part(.i.).fsub := false;
       end;
     for i := 1 to maxpart do begin
       for j := 1 to maxarea do begin
         area(.i,j.).name := '   ';
         area(.i,j.).count := 0;
         area(.i,j.).wrong := 0;
         area(.i,j.).correct := 0;
         area(.i,j.).flag := false;
       end;
     end;
     areas := 0;
         area(.1,1.).name := 'An Introduction';
         area(.1,2.).name := 'The Airplane and How It Flies';
         area(.1,3.).name := 'Cockpit: Instruments and Systems';
         area(.1,4.).name := 'Preflight Check';
         area(.1,5.).name := 'Starting The Airplane';
         area(.1,6.).name := 'Taxiing';
         area(.1,7.).name := 'Pretake-off or Cockpit Check';
         x := 0;
     for i := 1 to maxcat do begin
         problem(.i.).flag := false;
         problem(.i.).count := 0;
         problem(.i.).correct := 0;
         problem(.i.).wrong := 0;
         problem(.i.).easy := 0;
         problem(.i.).wasy := 0;
         for j := 1 to maxproblem do
             problem(.i.).comm(.j.) := '   ';
       end; (* for *)
     p:=0; n:=0; m:=0; comm := '   ';
     problem(.1.).name := ' Question about Procedures';
     problem(.2.).name := ' Question about Calculation';
     problem(.3.).name := ' Physical questions';
```

```pascal
end;
(*******************************************************************
 *                    Procedure Space                             *
 *        This procedure is called to space between lines.        *
 *******************************************************************)
procedure space( a:integer);
var
    i : integer;
begin
     for i := 1 to a do
          writeln;
end;
(*******************************************************************
 *                    Procedure Clearscreen                       *
 *        This procedure is called for easy going to the next page. *
 *******************************************************************)
procedure clearscreen;
begin
     space(2);
     writeln('     " press any key to continue " ');
     readln;
     page;
end;
procedure pilotinfo;
begin
    writeln('please type your number');
    readln(testnum);
    writeln(' your name ?');
    readln(name);
    writeln(' today date,(ex. 10 Apr 87)');
    readln(todaydate);
end;
procedure welcome;
begin
   space(5);
   writeln('***************************************************':66);
   writeln('*                                                 *':66);
   writeln('*                                                 *':66);
   writeln('*          WELCOME   TO   THE FIELD   OF   AVIATION *':66);
   writeln('*                                                 *':66);
   writeln('*                                                 *':66);
   writeln('***************************************************':66);
   space(1);
   pilotinfo;
   writeln(' "hit enter to continue"');
      readln; page;
end;
(*******************************************************************
 *                    Procedure Information          -            *
 *        This procedure provides information to the student of how *
 * the questions in each test is made and what will happen during test *
 *******************************************************************)
procedure information;
begin
   space(2);
   writeln('***************************************************':66);
   writeln('*                                                 *':66);
   writeln('*        In each test which you will work on :-    *':66);
   writeln('*                                                 *':66);
   writeln('*     1)  There are 3 sets of questions.           *':66);
   writeln('*     2)  Each set is different in level of difficulty *':66);
   writeln('*     3)  The most difficult set will be presented *':66);
   writeln('*          first and then the next two sets.       *':66);
   writeln('*     4)  You will be evaluated after completing each *':66);
   writeln('*          of the first two set of questions.      *':66);
   writeln('*     5)  If you are about average, all questions  *':66);
   writeln('*          will be presented, otherwise you do not *':66);
```

55

```pascal
            writeln('*          have to do all of the questions.          *':66);
            writeln('*                                                    *':66);
            writeln('*****************************************************':66);
         clearscreen;
      end;
   (*****************************************************************
    *                  Procedure Selected_book                     *
    *        This procedure determines the book that selected by the *
    *        student to provide questions from that book(concept).   *
    *****************************************************************)
   procedure selected_book(var con : boolean);
   var flag : boolean;
   begin
       flag := false;
       while not flag do begin
       readln(book);
       if book = '1' then begin
           b := 1;
           books(.b.).name := '" STUDENT PILOT FLIGHT MANUAL "';
           flag := true; con := true;
           books(.b.).fname := true;
          end
       else if book = '2' then begin
           b := 2;
           flag := true; con := true;
           books(.b.).name := ' " PRIVATE PILOT STUDY GUIDE "';
           books(.b.).fname := true;
          end
       else if book = '3' then begin
           b := 3;
           flag := true; con := true;
           books(.b.).name := '   EMPTY FILE !  ';
           books(.b.).fname := true;
          end
       else if (book = '0')  then begin
           flag := true;  con := false;
          end
       else begin
           writeln('  "only number please" ');
           flag := false;
         end;
       end;
   end;
   (*****************************************************************
    *                  Procedure Booklist                          *
    *        This procedure displays menu of the booklist for student *
    *        to select which book he/she want to test upon.          *
    *****************************************************************)
   procedure booklist(var con:boolean);
   begin
      space(2);
      writeln('*****************************************************':66);
      writeln('*                                                    *':66);
      writeln('*           PLEASE SELECT ONE OF THE FOLLOWING       *':66);
      writeln('*                                                    *':66);
      writeln('*           1.....STUDENT PILOTS FLIGHT MANUAL       *':66);
      writeln('*                                                    *':66);
      writeln('*           2.....PRIVATE PILOT STUDY GUIDE          *':66);
      writeln('*                                                    *':66);
      writeln('*           3.....INSTRUMENT FLIGHT RULES            *':66);
      writeln('*                                                    *':66);
      writeln('*           0.....for  " QUIT "  (exit program)      *':66);
      writeln('*                                                    *':66);
      writeln('*             (Type only number you want )           *':66);
      writeln('*                                                    *':66);
      writeln('*****************************************************':66);
      selected_book(con);
      writeln(':',book);
```

56

```pascal
end;
procedure firstheading2;
begin
    space(5);
    writeln('***********************************************************':66);
    writeln('*                                                         *':66);
    writeln('*              THE FOLLOWING ARE QUESTIONS                *':66);
    writeln('*                                                         *':66);
    writeln('*                         FROM                            *':66);
    writeln('*                                                         *':66);
    writeln('*            " PRIVATE PILOT STUDY GUIDE "                 *':66);
    writeln('*                                                         *':66);
    writeln('*                                                         *':66);
    writeln('***********************************************************':66);
    clearscreen;
end;
procedure firstheading1;
begin
    space(5);
    writeln('***********************************************************':66);
    writeln('*                                                         *':66);
    writeln('*              THE FOLLOWING ARE QUESTIONS                *':66);
    writeln('*                                                         *':66);
    writeln('*                         FROM                            *':66);
    writeln('*                                                         *':66);
    writeln('*          " STUDENT PILOT FLIGHT MANUAL "                *':66);
    writeln('*                                                         *':66);
    writeln('*                                                         *':66);
    writeln('***********************************************************':66);
    clearscreen;
end;

(***********************************************************************
 *                  Procedure Beginsession1                          *
 *         This procedure calls procedure booklist, information,and   *
 *         firstheading, passing boolean continue for begin questioning *
 *         or back to select another concept or quit.                *
 ***********************************************************************)
procedure beginsession1(var continue:boolean);
var
    fl, flag : boolean;
begin
    fl := true;
    booklist(continue);
    while fl do begin
        if book = '1' then begin page;
            information;
            firstheading1; fl:=false;
          end
        else if book = '2' then begin page;
            information;
            firstheading2;fl:=false;
          end
        else if book = '3' then begin page;
            fl := false; space(7);
            writeln('       THERE IS NO CONCEPT TO TEST HERE');
            space(4);
            clearscreen;
          end
        else if book = '0' then begin page;
            fl := false;
            continue := false;
          end;
    end;
end;
(***********************************************************************
 *                  Procedure Continuesession                        *
 *         This procedure displays screen to ask whether or not the   *
```

```
*        student want to continue testing on the same or different    *
*        concept.                                                      *
*****************************************************************************)
procedure continuesession(var cont:char;var select:boolean);
begin
    space(5);
    writeln('*********************************************************':66);
    writeln('*                                                        *':66);
    writeln('*                DO YOU WANT TO CONTINUE                 *':66);
    writeln('*                ON THE SELECTED CONCEPT?                *':66);
    writeln('*                                                        *':66);
    writeln('*        PRESS " ANY KEY "  to continue                  *':66);
    writeln('*                             on the selected concept    *':66);
    writeln('*                                                        *':66);
    writeln('*        PRESS " N "  to select another lesson           *':66);
    writeln('*                          or go to exit                 *':66);
    writeln('*                                                        *':66);
    writeln('*********************************************************':66);
    readln(cont);
    writeln(':',cont);
    page;
        if (cont ='n') or (cont = 'N') then begin
            select := false;
            next := true;
          end
      else begin
            select := true;
            next := true;
        end;
end;
  (***************************************************************************
  *                    Procedure Calculate1                                 *
  *         This procedure calculates probability of student ability after*
  * completed the critical set of questions. This procedure is called     *
  * by procedure display_question, values are passed in/out, for example*
  * number of correct answer, questions done, and incorrect answer.       *
  ***************************************************************************)
procedure calculate1(chap:integer;cor:integer;wro:integer;cou:integer;
                        var prob : real;var m1:integer;var r1:integer);
var
        pu,px,a,b,x,y,z,d,e : real;
        i,w1 : integer;
begin
        m1 := chapter(.chap.).crit;
        r1 := cor;
        w1 := wro;
        chapter(.chap.).rcrit := cor;
        chapter(.chap.).wcrit := wro;
        space(5);
        writeln('    number question in critical set  = ', m1:3);
        writeln('    number correct answers    = ',r1:3);
        writeln('    number incorrect answers   = ',w1:3);
        (* calculate P(A1/BU) *)
        pu := uninformed;  px := expert;
         a := lowweight;   b := highweight;
        i := 1; x := 1;
        if ((cou-cor) <> 0) then begin
            for i := 1 to (cou-cor) do
                x := x * (1-pu);
          end (*if*)
        else   x := 1;
        y := 1;
        if (cor <> 0) then begin
            for i := 1 to cor do
                y := y * pu;
          end (*if*)
        else   y := 1;
        z := 1;
        if (cou <> cor) then begin
```

```
                  for i := cou downto (cor+1) do
                       z := i * z;
                  end (*if*)
              else   z := 1;
              d := z*x*y;           (* value P(A1/BU) *)
       (* calculate P(A1/BX) *)
              i := 1; x := 1;
              if ((cou-cor) <> 0) then begin
                  for i := 1 to (cou-cor) do
                       x := x * (1-px);
                  end (*if*)
              else   x := 1;
              y := 1;
              if (cor <> 0) then begin
                  for i := 1 to cor do
                       y := y * px;
                  end (*if*)
              else   y := 1;
              z := 1;
              if (cou <> cor) then begin
                  for i := cou downto (cor+1) do
                       z := i * z;
                  end (*if*)
              else   z := 1;
              e := z*x*y;   (* value P(A1/BX) *)
         (* find P(BU) *)
              x := a/(a+b);
         (* find P(BX) *)
              y := b/(a+b);
         (* find P(BU/A1) *)
              x := (x*d)/((x*d)+(y*e));
         (* find P(BX/A1) *)
              prob := (1 - x);
              writeln (' The probability of your ability is   ', prob:8:3);
              chapter(.chap.).rcrit := cor;
              chapter(.chap.).wcrit := wro;
end;  (*  end procedure calculate1 *)
  (*****************************************************************************
  *                   Procedure Calculate2                                    *
  *        This procedure calculates probability of student ability after*
  * completed the second set of questions. This procedure is called      *
  * by procedure display_question, values are passed in/out, for example*
  * number of correct answer, questions done, and incorrect answer.      *
  ****************************************************************************)
procedure calculate2(chap:integer;cor:integer;wro:integer;cou:integer;
                          var prob : real;var m2:integer;var r2:integer);
var    pu,px,a,b,x,y,z,d,e : real;
       i,w2,m3 : integer;
begin
       m3 := chapter(.chap.).comp;
       m2 := cou - chapter(.chap.).crit;
       r2 := cor - chapter(.chap.).rcrit;
       w2 := wro - chapter(.chap.).wcrit;
       chapter(.chap.).rcomp := r2;
       chapter(.chap.).wcomp := w2;
       space(5);
       writeln('    number question in complementary set  = ',m3:3);
       writeln('    number correct answers  = ',r2:3);
       writeln('    number incorrect answers   = ',w2:3);
       (* calculate P(A2/BU) *)
       pu := uninformed;  px := expert;
        a := lowweight;   b := highweight;
       i := 1; x := 1;
       if ((m2-r2) <> 0) then begin
           for i := 1 to (m2-r2) do
                x := x * (1-pu);
           end (*if*)
       else   x := 1;
       y := 1;
```

59

```pascal
            if (r2 <> 0) then begin
                for i := 1 to r2 do
                    y := y * pu;
                end (*if*)
            else   y := 1;
            z := 1;
            if (m2 <> r2) then begin
                for i := m2 downto (r2+1) do
                    z := i * z;
                end (*if*)
            else   z := 1;
            d := z*x*y;          (* value P(A1/BU) *)
        (* calculate P(A1/BX) *)
            i := 1; x := 1;
            if ((m2-r2) <> 0) then begin
                for i := 1 to (m2-r2) do
                    x := x * (1-px);
                end (*if*)
            else   x := 1;
            y := 1;
            if (r2 <> 0) then begin
                for i := 1 to r2 do
                    y := y * px;
                end (*if*)
            else   y := 1;
            z := 1;
            if (m2 <> r2) then begin
                for i := m2 downto (r2+1) do
                    z := i * z;
                end (*if*)
            else   z := 1;
            e := z*x*y;    (* value P(A1/BX) *)
        (* find P(BU) *)
            x := a/(a+b);
        (* find P(BX) *)
            y := b/(a+b);
        (* find P(BU/A1) *)
            x := (x*d)/((x*d)+(y*e));
        (* find P(BX/A1) *)
            prob := (1 - x);
            writeln (' The probability of your ability is  ', prob:8:3);
            chapter(.chap.).rcomp := cor - chapter(.chap.).rcrit;
            chapter(.chap.).wcomp := wro - chapter(.chap.).wcrit;
end;  (*  end procedure calculate2 *)
 (*****************************************************************
 *                   Procedure Check_answer                      *
 *        This procedure checks the student's answer. If the answer   *
 *    is invalid, the error message will be given to the student to   *
 *    type in the valid answer again.                            *
 *****************************************************************)
procedure check_answer(var c : char);
var
     flag : boolean;
begin
     flag := true;
     while flag do begin
        if (c='1')or(c='2')or(c='3')or(c='4') then
            flag := false
        else begin
            writeln('type again  1..to..4 please');
            readln(c);
            flag := true; end;
      end;
end;
 (*****************************************************************
 *                   Procedure Alter_cats                        *
 *        This procedure will change the categories of question  *
 *    which is read from text file to integer type to be used as *
 *    the value in array of questions categories.                *
```

60

```
    ***********************************************************)
procedure alter_cats;
begin
    if cats = 'p' then cat := 1;
    if cats = 'c' then cat := 2;
    if cats = 's' then cat := 3;
    problem(.cat.).flag := true;
    problem(.cat.).count := problem(.cat.).count + 1;
end;
   (***********************************************************
   *                 Procedure Input_cat_comm                *
   *       This procedure will read any comment of the question *
   *    for every wrong answer made by student and then put each*
   *    comment in array of category which that questions belongs. *
   ***********************************************************)
procedure input_cat_comm;
begin
    if cat = 1 then begin
       p := p +1;
       problem(.cat.).comm(.p.) := comm;
       problem(.cat.).wrong := problem(.cat.).wrong + 1;
       end
    else if cat = 2 then begin
       m := m +1;
       problem(.cat.).comm(.m.) := comm;
       problem(.cat.).wrong := problem(.cat.).wrong + 1;
       end
    else if cat = 3 then begin
       n := n +1;
       problem(.cat.).comm(.n.) := comm;
       problem(.cat.).wrong := problem(.cat.).wrong + 1;
       end;
end;
   (***********************************************************
   *                 Procedure Compare_answer                *
   *       This procedure calls procedure check_answer to determine *
   *    student's answer is correct or incorrect and , count number of *
   *    questions done, correct and incorrect answer and also record *
   *    the comment of every wrong answer in order to provide suggestions *
   *    and recommendations in the final conclusions.        *
   ***********************************************************)
procedure compare_answer(var count:integer;var correct:integer;
                    var wrong:integer;var a:integer;var filename:text;
                    var chapt:integer);
var
    symbol,ease: char;i:integer;
begin
    readln(answer); check_answer(answer);    (* from screen *)
    readln(filename,correctans);
    read(filename,areas); read(filename,cats);
    readln(filename,ease);
    alter_cats;
    if ease = 'e' then problem(.cat.).easy := problem(.cat.).easy+1;
    area(.chapt,areas.).flag := true;
    writeln('    Your answer is   ',answer);
      if answer = correctans then begin
            correct := correct + 1;
            count := count + 1;
            chapter(.chapt.).correct := chapter(.chapt.).correct +1;
            chapter(.chapt.).count := chapter(.chapt.).count +1;
            area(.chapt,areas.).count := area(.chapt,areas.).count + 1;
            area(.chapt,areas.).correct := area(.chapt,areas.).correct+1;
            problem(.cat.).correct := problem(.cat.).correct + 1;
            writeln('  _____');
            writeln('  *   You are correct    *');
            writeln('  _____');
            readln(filename,question);
            clearscreen;
        end(* if *)
```

61

```pascal
        else begin
            wrong := wrong + 1;
            count := count + 1;
            chapter(.chapt.).wrong := chapter(.chapt.).wrong +1;
            chapter(.chapt.).count := chapter(.chapt.).count +1;
            area(.chapt,areas.).count := area(.chapt,areas.).count + 1;
            area(.chapt,areas.).wrong := area(.chapt,areas.).wrong + 1;
            writeln('                                  ');
            writeln(' * You are wrong, answer is ',correctans:3,'*':2);
            writeln('                                  ');
            if ease = 'e' then
                problem(.cat.).wasy := problem(.cat.).wasy+1;
            read(filename,symbol);
            clearscreen;
            readln(filename,comm);
            input_cat_comm;
            comment(.chapt,a.).pages := comm;
            numcomment(.chapt.) := a;
                a := a+1;
        end; (* else *)
  end;(* procedure *)
  (******************************************************************
  *                 Procedure Decision                            *
  *     This procedure compares the probability of student ability *
  *   passed from procedure calculate1&2 to the cut-off value .4,.9 *
  *   then make decision about student's knowledge in that test. This *
  *   procedure is called by procedure display_question.          *
  ******************************************************************)
procedure decision(p:real;var sel:boolean;var chapt:integer;
                          var flag:boolean);
begin
    if (p = alpha) or (p > alpha) then begin
        if flag then begin
            rec(.chapt.).av := '" YOU ARE ABOUT AVERAGE IN THIS PART  "';
            rec(.chapt.).flag3 := true;
        end
        else begin
            rec(.chapt.).ex:= '" YOU ARE ABOVE AVERAGE, EXCELLENT! "';
            rec(.chapt.).flag1 := true;
        end;
    end(* if *)
    else if (p = beta) or (p < beta) then begin
        rec(.chapt.).poor :='" YOU ARE BELOW AVERAGE IN THIS PART "';
        rec(.chapt.).flag2 := true;
    end(* else if *)
    else if (p > beta) and (p < alpha) then begin
        rec(.chapt.).av := '" YOU ARE ABOUT AVERAGE IN THIS PART  "';
        rec(.chapt.).flag3 := true;
    end;
end;
  (******************************************************************
  *                 Procedure Make_decision                       *
  *     This procedure compares the probability of student ability *
  *   passed from procedure calculate1&2 to the cut-off values .4,.9 *
  *   then make decision to go on next questions set or not.     If *
  *   the probability is in between the cut-off values then present *
  *   next set of question otherwise stop questioning the student. *
  ******************************************************************)
procedure make_decision(p:real;var sel:boolean;var chapt:integer;
                          var flag:boolean);
begin
    if (p = alpha) or (p > alpha) then begin
        space(7);
        writeln('**********************************************':63);
        writeln('*                                          *':63);
        writeln('*      YOUR KNOWLEDGE IN THE PREVIOUS LESSON   *':63);
        writeln('*             IS SATISFACTORY              *':63);
        writeln('*        YOU ARE FINISHED WITH THIS LESSON    *':63);
        writeln('*                                          *':63);
```

```pascal
            writeln('***************************************************':63);
            space(3);
            clearscreen;
            sel := false;
          end
        else if (p = beta) or (p < beta) then begin
          space(5);
          writeln('*********************************************':63);
          writeln('*                                           *':63);
          writeln('*          YOU ARE WEAK FOR THIS LESSON      *':63);
          writeln('*                                           *':63);
       writeln('*':19,'STUDY':7,part(.chapt.).subname:28,'AGAIN':7,'*':2);
          writeln('*                                           *':63);
          writeln('*        YOU ARE FINISHED WITH THIS LESSON   *':63);
          writeln('*                                           *':63);
          writeln('*********************************************':63);
          space(2);
          clearscreen;
             sel := false;
          end(* else if *)
        else if (p > beta) and (p < alpha) then begin
          space(7);
          writeln('************************************************':66);
          writeln('*                                              *':66);
          writeln('*           YOU ARE ABOUT AVERAGE SO FAR       *':66);
          writeln('* WE WILL CONTINUE ON THE REST OF THE QUESTIONS *':66);
          writeln('*                                              *':66);
          writeln('************************************************':66);
          space(3);
          clearscreen;
          sel := true;
        end;
end;

procedure suggesthead;
begin
      writeln;
      writeln('_____');
      writeln(' *   YOUR PROBLEMS AND SUGGESTIONS FOR IMPROVING    *');
      writeln('_____');
      writeln;
end;
  (************************************************************************
   *                      Procedure Evaluate                            *
   *        This procedure is called by procedure display_question.      *
   *   At the end of the first two set of questions, this procedure will *
   *   be called to provide information to the student of the previous   *
   *   performances then procedure make_decision will be called next.    *
   **********************************************************************)
procedure evaluate(var chap:integer);
var
      j,i : integer;
begin
      writeln('_____');
      space(1);
      if(chapter(.chap.).count<>chapter(.chap.).block) then
      writeln(' You have not yet completed all the questions.':50)
      else
      writeln(' You have completed all the questions at this time.':54);
      space(1);
   writeln(' You will now be evaluated on what you have done so far.'
                                                                      :55);
      space(1);
      writeln('THERE ARE':17,chapter(.chap.).block:3,'  QUESTIONS.');
      write(chapter(.chap.).crit:12);
      writeln('critical questions.':21);
      write(chapter(.chap.).comp:12);
      writeln('complementary questions.':26);
      write(chapter(.chap.).sup:12);
      writeln('supplementary questions.':26);
```

63

```pascal
        space(1);
        if(chapter(.chap.).count=chapter(.chap.).crit) then
           writeln('you have answered only the critical questions set.':55)
        else if(chapter(.chap.).count=chapter(.chap.).crit +
                                     chapter(.chap.).comp) then
           writeln('you have answered both critical and complememtary sets'
                                                                              :61)
        else if(chapter(.chap.).count=chapter(.chap.).block) then
             writeln('you have answered all the questions': 40);
        writeln('correct answers    =':30,chapter(.chap.).correct:5);
        writeln('incorrect answers    =':32,chapter(.chap.).wrong:5);
        space(1);
        if (rec(.chap.).flag1 = true)  then begin
             writeln(rec(.chap.).ex:46);
           end
        else if (rec(.chap.).flag2 = true) then begin
             writeln(rec(.chap.).poor:45);
           end
        else if (rec(.chap.).flag3 = true) then begin
             writeln(rec(.chap.).av:45);
           end;
        writeln;
        if (problem(.1.).wasy > 0) or (problem(.2.).wasy > 0) or
             (problem(.3.).wasy > 0) then
             writeln(' * * Be careful on easy questions! **');
             if (problem(.1.).wasy > 0) then begin
                  write(' You have made',problem(.1.).wasy:2);
                  writeln(' mistakes about procedures.');
                end;
             if (problem(.2.).wasy > 0) then begin
                  write(' You have made',problem(.2.).wasy:2);
                  writeln(' mistakes about calculation.');
                end;
             if (problem(.3.).wasy > 0) then begin
                  write(' You have made',problem(.3.).wasy:2);
                  writeln(' mistakes about physical question.');
                end;
        writeln('_____');
  end;
  (*******************************************************************
  *                    Procedure One_chapter_conclusion              *
  *      This procedure is called by procedure ask_conclusion at the *
  *   end of each test, whether the student want to see the results  *
  *   of the previous test or not.                                   *
  *******************************************************************)
procedure one_chapter_conclusion(var chap:integer);
var
     j,i,k : integer;
begin
     space(1);
     for i := 1 to maxbook do begin
          if (b = i) then begin
               writeln(books(.i.).name : 40);
               space(1);
               for j := 1 to maxpart do begin
                    if (chap = j) then
                         writeln(part(.j.).subname :38);
                  end;(* for*)
             end; (* if*)
        end; (*for*)
     space(1);
     writeln('THERE ARE':17,chapter(.chap.).block:3,'  QUESTIONS ');
     write(chapter(.chap.).crit:12);
     writeln('critical questions':21);
     write(chapter(.chap.).comp:12);
     writeln('complementary questions':26);
     write(chapter(.chap.).sup:12);
     writeln('supplementary questions':26);
     space(1);
```

```pascal
        if(chapter(.chap.).count=chapter(.chap.).crit) then
            writeln('you have completed only the critical questions':43)
        else if(chapter(.chap.).count=chapter(.chap.).crit +
                                        chapter(.chap.).comp) then
          writeln('you have completed both critical and complememtary':58)
        else if(chapter(.chap.).count=chapter(.chap.).block) then
            writeln('you have answered all the questions': 45);
      writeln('Total correct answers  =':36,chapter(.chap.).correct:5);
      writeln('Total incorrect answers  =':38,chapter(.chap.).wrong:5);
      space(1);
      if (rec(.chap.).flag1 = true) then
            writeln(rec(.chap.).ex:46)
      else if (rec(.chap.).flag2 = true) then
            writeln(rec(.chap.).poor:45)
      else if (rec(.chap.).flag3 = true) then
            writeln(rec(.chap.).av:45);
      writeln;
      writeln('Following is a report of your performance':50);
      writeln;
      for i := 1 to maxarea do begin
          if (area(.1,i.).flag) then begin
              writeln(area(.1,i.).name);
              write('Pts =':10,area(.1,i.).correct:3,',','Total =':10,
              area(.1,i.).count:3,',','Percent Grade =':17);
              x := ((area(.1,i.).correct/area(.1,i.).count)*100 );
              writeln(x:8:2);
              writeln;
          end; (*if*)
      end;(*for*)
      suggesthead;
      for i := 1 to maxcat do begin
          if (problem(.i.).flag) then begin
              writeln(problem(.i.).name);
              write('Pts =':10,problem(.i.).correct:3,',','Total =':10,
              problem(.i.).count:3,',','Percent Grade =':17);
              x:=(problem(.i.).correct/problem(.i.).count)*100;
              writeln(x:8:2);
              if (problem(.i.).wrong <> 0) then begin
                  writeln(' You should review as follows':32);
                  for k := 1 to problem(.i.).wrong do begin
                      write(k:5,')');
                      writeln(problem(.i.).comm(.k.));
                  end; (*for*)
                  writeln;
                  if x < 50 then
                      writeln('** Study more! You are weak here. **':40);
                  if problem(.i.).wasy > 0 then begin
                      writeln('   ** Be careful on easy question! **');
                      write('    You missed', problem(.i.).wasy:3);
                      write(' of them in the total of');
                      writeln(problem(.i.).easy:3,'.');
                  end;
                  writeln('_____')
              end; (*if*)
              writeln;
          end; (*if*)
      end;(*for*)
  end;
  (****************************************************************
  *                   Procedure Display_question               *
  *    This procedure calls procedure calculate1&2, decision,   *
  *  evaluate, and make_decision. Main activities are done in this *
  *  procedure, and is called by procedure select_chapter. Questions *
  *  will be read from a file and display through the screen.    *
  ****************************************************************)
  procedure display_question(var count:integer;var correct:integer;
                             var wrong:integer;var filename:text;
                             var chapt:integer;var flag:boolean);
    var
```

```pascal
        sel : boolean;
        i,m1,r1,w1,n1,m2,r2,w2,n2 : integer;
        p : real;
begin
        flag := false;
        m1 := 0; r1 := 0; w1 := 0;
        m2 := 0; r2 := 0; w2 := 0;
        n1 := chapter(.chapt.).crit;
        n2 := chapter(.chapt.).comp;
        count := 0;
        correct := 0;
        wrong := 0;
           i := 1;
        sel := true;
        while sel and not eof(filename) do
              begin
                   read(filename,symbol);
                   if symbol = '' then
                       begin
                           sel := false;
                           readln(filename,question);
                       end    (* if *)
                   else
                       begin
                           space(1);
                           while not (symbol = '$')  do
                               begin
                                     readln(filename, question);
                                     write(symbol); writeln(question);
                                     read(filename, symbol);
                                end;(* while *)
                                readln(filename,question);
                                write(' ');  writeln(question);
                                space(1);
                                compare_answer(count,correct,wrong,i,filename,
                                               chapt);
                        end;   (* else*)
                        if (count = n1) then begin
                            calculate1(chapt,correct,wrong,count,p,m1,r1);
                            clearscreen;
                            decision(p,sel,chapt,flag);
                            evaluate(chapt);
                            clearscreen;
                            make_decision(p,sel,chapt,flag);
                           end (*if*)
                        else if (count > n1) and (count = n1+n2) then
                           begin
                            flag := true;
                            calculate2(chapt,correct,wrong,count,p,m2,r2);
                            clearscreen;
                            decision(p,sel,chapt,flag);
                            evaluate(chapt);
                            clearscreen;
                            make_decision(p,sel,chapt,flag);
                           end   (*else*)
        end;(* outer while *)
end; (* procedure *)
 (****************************************************************
 *                    Procedure Storedata                      *
 *       This procedure is called by procedure process to store  *
 *    the datas of what chapter student have done, number of questions, *
 *    number correct and incorrect answer in each chapter.      *
 ****************************************************************)
procedure storedata(var c:integer;var cor:integer;var w:integer;
                        chapt:integer);
var
      i : integer;
begin
      for i := 1 to maxchap do begin
```

```pascal
            if (chapter(.i.).num = chapt ) then begin
                chapter(.i.).count := c;
                chapter(.i.).correct := cor;
                chapter(.i.).wrong := w;
            end; (* if *)
        end; (* for *)
end; (* procedure *)
  (**********************************************************************
  *                    Procedure Readchapter                          *
  *      This procedure is called by procedure what_chapter1&2 to     *
  *   input which chapter student want to test upon and the data is   *
  *   passed out to some other procedures during session.             *
  **********************************************************************)
procedure readchapter(var chap : integer);
var
    flag:boolean;
    c: char;
begin
    flag := false;
    while not flag do begin
        readln(c);
        if c = '1' then begin chap := 1; flag := true;end
        else if c = '2' then begin chap := 2; flag := true;end
        else if c = '3' then begin chap := 3; flag := true;end
        else if c = '0' then begin chap := 0; flag := true;end
        else begin
            writeln(' "type again only number please " ');
            flag := false;
        end; (*else*)
    end; (* while*)
end; (* proc*)

procedure what_chapter2(var chap : integer);
begin
    space(5);
    writeln('**************************************************':66);
    writeln('*                                                *':66);
    writeln('*           " PRIVATE PILOT STUDY GUIDE "         *':66);
    writeln('*              SELECT ONE OF THE FOLLOWING        *':66);
    writeln('*                                                *':66);
    writeln('*          TYPE  1..... for NAVIGATION            *':66);
    writeln('*          TYPE  2..... for RADIO  NAVIGATION     *':66);
    writeln('*          TYPE  3......for NIGHT FLYING          *':66);
    writeln('*          TYPE  0..... for SELECTING ANOTHER BOOK*':66);
    writeln('*                                                *':66);
    writeln('**************************************************':66);
    readchapter(chap);
    writeln(':',chap:2);
end;

procedure what_chapter1(var chap : integer);
begin
    space(5);
    writeln('**************************************************':66);
    writeln('*                                                *':66);
    writeln('*           " STUDENT PILOT FLIGHT MANUAL "       *':66);
    writeln('*              SELECT ONE OF THE FOLLOWING        *':66);
    writeln('*                                                *':66);
    writeln('*          TYPE  1..... for  PART - ONE           *':66);
    writeln('*          TYPE  2..... for  PART - TWO           *':66);
    writeln('*          TYPE  3......for  PART - THREE         *':66);
    writeln('*       TYPE  0..... for  SELECTING ANOTHER CONCEPT*':66);
    writeln('*                                                *':66);
    writeln('**************************************************':66);
    readchapter(chap);
    writeln(':',chap:2);
end;

procedure begin_chapter(chap:integer);
begin
    space(7);
```

```pascal
        writeln('********************************************':60);
        writeln('*                                          *':60);
        writeln('*   THIS IS THE BEGINNING OF THE LESSON    *':60);
        writeln('*                                          *':60);
        writeln('********************************************':60);
        space(5);
        clearscreen;
end;
  (************************************************************************
   *                    Procedure Input_question                        *
   *        This procedure is called by procedure process. After the     *
   *    student selected any chapter to test then this procedure will    *
   *    determine how many questions in each test and also in each set   *
   *    of questions for the system to use these informations later.     *
   ************************************************************************)
procedure input_question(chap:integer;var filename:text);
var
    q : integer;
begin
    readln(filename,q);
    chapter(.chap.).block := q;
    readln(filename,q);
    chapter(.chap.).crit := q;
    readln(filename,q);
    chapter(.chap.).comp := q;
    readln(filename,q);
    chapter(.chap.).sup := q;
end;
procedure end_chapter(chap:integer);
begin
    space(7);
    writeln('**********************************':54);
    writeln('*                                *':54);
    writeln('*   THIS IS THE END OF THE LESSON  *':54);
    writeln('*                                *':54);
    writeln('**********************************':54);
    space(5);
    clearscreen;
end;


procedure ask_conclusion(var chap:integer);
var
    ch : char;
    flag : boolean;
begin
    flag := true;
    space(7);
    writeln('****************************************************':66);
    writeln('*                                                *':66);
    writeln('*              DO YOU WANT TO SEE THE RESULTS     *':66);
    writeln('*                                                *':66);
    writeln('*             YES     PRESS " ANY KEY "          *':66);
    writeln('*             NO      PRESS " N "                *':66);
    writeln('*                                                *':66);
    writeln('****************************************************':66);
    readln(ch);
    writeln(':',ch);
    if (ch = 'n') or (ch = 'N') then flag := false
    else begin
        page;
        one_chapter_conclusion(chap);
        clearscreen;
        end;
end;
  (************************************************************************
   *                    Procedure Process                               *
   *        This procedure is called by procedure selectchapter. Five    *
   *    procedure are called by process for the purpose of eliminating   *
```

68

```
 *    redundancy in procedure selectchapter.                              *
  ****************************************************************************)
procedure process(var cou :integer;var cor:integer;var wr:integer;
                    var chap:integer;var sel:boolean;var flag:boolean;
                    var filename:text);
var cont : char;
begin
     input_question(chap,filename);
     display_question(cou,cor,wr,filename,chap,flag);
     storedata(cou,cor,wr,chap);
     end_chapter(chap);
     ask_conclusion(chap);
     page;
     continuesession(cont,sel);
end;
  (****************************************************************************
  *                    Procedure Selectchapter                              *
  *       This procedure calls procedure process after determined which *
  *    chapter is needed by the student to work on. It is called by       *
  *    procedure open_close_session after knowing the student want to     *
  *    begin the test. This procedure is designed to work only two        *
  *    concepts and only three chapter in a concept.                      *
  ****************************************************************************)
procedure selectchapter(var select : boolean;var count:integer;
                          var correct:integer;var wrong:integer);
var
    chap : integer;   (* number of chapter *)
    cont : char;      (* read y/n to continue next chapter *)
    flag : boolean;
begin
    flag := false;
    chap := 0;
    select := true;
    while select do begin
        case book of
          '0' : select := false;
          '1' : begin
                    what_chapter1(chap);
                    page;
                    case chap of
                    0 :  select := false;

                    1 : begin
                            begin_chapter(chap);
                    part(.chap.).subname := '"PART - ONE" (Before Flight)';
                            part(.chap.).fsub := true;
                            reset(file1,'file1 text');
                            process(count,correct,wrong,chap,select,flag,file1)
                        end;
                    2 : begin space(7);
                            writeln('  FILE IS EMPTY,  NO QUESTION !');
                            writeln('         ..NEED TO BE FILLED IN.. ');
                            clearscreen;
                            continuesession(cont,select);
                        end;
                    3 : begin space(7);
                            writeln('  FILE IS EMPTY,  NO QUESTION !');
                            writeln('         ..NEED TO BE FILLED IN.. ');
                            clearscreen;
                            continuesession(cont,select);
                        end;
                  end;(* case *) end;
          '2' : begin
                    what_chapter2(chap);
                    page;
                    case chap of
                    0 :  select := false;
                    1 :  begin
                            writeln('  FILE IS EMPTY,  NO QUESTION !');
                            clearscreen;
```

```
                          continuesession(cont,select);
                     end;
                2 : begin space(7);
                          writeln('  FILE IS EMPTY,  NO QUESTION !');
                          clearscreen;
                          continuesession(cont,select);
                     end;
                3 : begin space(7);
                          writeln('  FILE IS EMPTY,  NO QUESTION !');
                          clearscreen;
                          continuesession(cont,select);
                       end;
              end; (* case *) end;
          '3' : select:= false;
        end; (* case *)
     end; (* while *)
end; (* proc *)
procedure goodbye;
begin
     space(8);
     writeln('**********************************':54);
     writeln('*                                *':54);
     writeln('*      HOPE TO SEE YOU AGAIN      *':54);
     writeln('*                                *':54);
     writeln('**********************************':54);
     space(4);
     clearscreen;
end;
  (***************************************************************************
  *                      Procedure Inputfile                              *
  *      This procedure updates the student's performance in the          *
  *  historical text file and also compare student's performance          *
  *  between his last test and the present test. It is called by          *
  *  Procedure Conclusions at the end of the session.                     *
  ***************************************************************************)
procedure inputfile(y:real);
var found : boolean;
begin
    reset(pilotf,'pilotf data');
    rewrite(tempfile,'tempfile data');
    found := false;
    repeat
        read(pilotf,pilotnum);
        if (testnum = pilotnum) then begin
            write(tempfile,pilotnum:10);
            read(pilotf,pilotname);
            write(tempfile,pilotname:20);
            read(pilotf,pilotdate); readln(pilotf,pilotgrade);
            writeln;
            writeln('     Your last test was on ',pilotdate,'.');
            writeln('     And the grade was ', pilotgrade:8:3,'.');
            if pilotgrade > y then begin
                writeln(' Your grade today was',y:8:2,'.');
                writeln(' You are weaker than last time,"Study harder"!');
              end
            else if pilotgrade < y then begin
                writeln(' Your grade today was',y:8:2,'.');
                writeln(' Good, you are improving!');
              end
             else
                writeln(' Your grade today was',y:8:2,',','  SAME');
            pilotgrade := y;
            pilotdate := todaydate;
           write(tempfile,pilotdate:15); writeln(tempfile,pilotgrade:8:2);
            found := true;
          end(*if*)
        else begin
            write(tempfile,pilotnum:10);
            read(pilotf, pilotname);
```

70

```pascal
                        write(tempfile,pilotname:20);
                        read(pilotf,pilotdate); readln(pilotf,pilotgrade);
                        write(tempfile,pilotdate:15);
                        writeln(tempfile,pilotgrade:8:2);
                    end;(*else*)
            until eof(pilotf);
            if not found then
                writeln(tempfile, testnum :10, name:20, todaydate:15, y:8:2);
            reset(tempfile,'tempfile data');
            rewrite(pilotf,'pilotf data');
            repeat
                    read(tempfile,pilotnum);
                    read(tempfile,pilotname);
                    read(tempfile,pilotdate); readln(tempfile,pilotgrade);
                    write(pilotf,pilotnum:10); write(pilotf,pilotname:20);
                    write(pilotf,pilotdate:15);
                    writeln(pilotf,pilotgrade:8:2);
            until eof(tempfile);
end;
procedure heading_conclusion;
begin
        space(6);
        writeln('*****************************************':57);
        writeln('*                                       *':57);
        writeln('*            FINAL SECTION:             *':57);
        writeln('*        SCORES AND COMMENTARY          *':57);
        writeln('*                                       *':57);
        writeln('*****************************************':57);
        space(4);
        clearscreen;
end;
    (****************************************************************************
     *                    Procedure Conclusion                                 *
     *         This procedure will be called at the end of the session.         *
     *   It provides all informations of what have been done by the             *
     *   student and also provides student's difficulties and details           *
     *   recommendations in each test.                                          *
     ****************************************************************************)
procedure conclusion;
var
        i,j,k : integer;  (* index *)
        g : real;(* grade *)
begin
        space(2);
        writeln('======================================================');
        writeln('                 SCORES AND COMMENTS                  ');
        writeln('======================================================');
        for j := 1 to maxpart do begin
                while (part(.j.).fsub) do begin
                    if (j = 1) or ( j = 2) then begin
                        space(1);
                        writeln(books(.1.).name:40);
                        space(1);
                        writeln(part(.j.).subname :38); space(1);
                    end (*if*)
                    else if (j = 3) or ( j = 4) then begin
                        writeln(books(.2.).name:45); space(1);
                        writeln(part(.j.).subname :37); space(1);
                    end;(*else*)
                    writeln('THERE ARE':17,chapter(.j.).block:3,'  QUESTIONS.');
                    write(chapter(.j.).crit:12);
                    writeln('critical questions.':21);
                    write(chapter(.j.).comp:12);
                    writeln('complementary questions.':26);
                    write(chapter(.j.).sup:12);
                    writeln('supplementary questions.':26);
                    space(1);
                    if(chapter(.j.).count=chapter(.j.).crit) then
                    writeln('you have answered only the critical questions set'
```

71

```pascal
        else if(chapter(.j.).count=chapter(.j.).crit +
                              chapter(.j.).comp) then
          writeln('you have answered both critical and complememtary'

        else if(chapter(.j.).count=chapter(.j.).block) then
          writeln('you have answered all the questions': 45);
      writeln('Total correct answers  =':36,chapter(.j.).correct:5);
      writeln('Total incorrect answers  =':38,chapter(.j.).wrong:5);
        writeln;
        if (rec(.j.).flag1 = true) then
           writeln(rec(.j.).ex:46)
        else if (rec(.j.).flag2 = true) then
           writeln(rec(.j.).poor:45)
        else if (rec(.j.).flag3 = true) then
           writeln(rec(.j.).av:45);
      g := (chapter(.j.).correct/chapter(.j.).count)*100;
      inputfile(g);  (* compare to history *)
      space(2);
    writeln('Following is a report of your performance':50);
    writeln;
    for i := 1 to maxarea do begin
        if (area(.j,i.).flag) then begin
           writeln(area(.j,i.).name);
           write('Pts =':10,area(.j,i.).correct:3,' ',' ','Total =':10,
              area(.j,i.).count:3,' ',' ','Percent Grade =':17);
           x := ((area(.j,i.).correct/area(.j,i.).count)*100 );
           writeln(x:8:2);
           writeln;
        end; (*if*)
     end;(*for*)
    suggesthead;
    for i := 1 to maxcat do begin
        if (problem(.i.).flag) then begin
           writeln(problem(.i.).name);
           write('Pts =':10,problem(.i.).correct:3,' ',' ','Total =':10,
           problem(.i.).count:3,' ',' ','Percent Grade =':17);
           x:=(problem(.i.).correct/problem(.i.).count)*100;
           writeln(x:8:2);
           if (problem(.i.).wrong <> 0) then begin
              writeln(' You should review as follows':32);
              for k := 1 to problem(.i.).wrong do begin
                 write(k:5,')');
                 writeln(problem(.i.).comm(.k.));
               end; (*for*)
              writeln;
              if x < 50 then
                 writeln('** Study more! You are weak here. **':40);
              if problem(.i.).wasy > 0 then begin
                 writeln('    ** Be careful on easy question! **');
                 write('    You missed', problem(.i.).wasy:3);
                 write(' of them in the total of');
                 writeln(problem(.i.).easy:3,'.');
               end;
               writeln('_____')
            end; (*if*)
           writeln;
         end; (*if*)
      end;(*for*)
     clearscreen;
        part(.j.).fsub := false;
     end;
    end; (* else *)
end; (* procedure *)
 (********************************************************************
 *                  Procedure open_close_session                    *
 *     This procedure receives the boolean flag from procedure      *
 *  beginsession1, if true then the session will begin otherwise    *
 *  it will set program flag to false and exit the program.         *
```

```
   ***********************************************************)
procedure open_close_session(var continue:boolean;var select:boolean;
               var count:integer;var correct:integer;var wrong:integer);
var y : real;
begin
     if continue = false then begin
         next := false;
         heading_conclusion;
         conclusion;
      end
      else    selectchapter(select,count,correct,wrong);
end; (* procedure *)
        (*=========================================*)
        (*                MAIN  PROGRAM*)
        (*=========================================*)
begin
page;
    welcome;
    initializevalue;
    repeat
       beginsession1(continue);
       open_close_session(continue,choose,numq,right,wrongq);
    until next = false;
    goodbye;
end.  (* end of program *)
```

LIST OF REFERENCES

1. Taracad R. Sivasankaran and Tung Bui, *A Bayesian Diagnostic Model for Intelligent CAI Systems*, Naval Postgraduate School, June 1986.

2. William.K. Kershner, *Student Pilot's Flight Manual*, Iowa State University press, Fifth edition, 1979.

3. Pitts, Gerald N. and Eggen, Maurice L., *CAI Course-ware Development Through Interactive Prompting*, AEDS Twenty-first Annual Convention Proceedings, p. 242, May 1983.

4. Korbak, Michael Jr., *Computer-Managed Instruction: Theory, Application, and Some Key*, implementation Issues, p. 19, Master's Thesis, Naval Postgraduate School, Monterey, California, March 1984.

5. Korbak, Michael Jr., *Computer-Managed Instruction: Theory, Application, and Some Key*, implementation Issues, pp. 19-20, Master's Thesis, Naval Postgraduate School, Monterey, California, March 1984.

6. Office of Naval Research Tech Memo No.17, *Current Research Development in Computer Assisted Instruction*, p. 2, by D.N. Hansen.

7. Baughman, Jo Ann, *Developing Instruction Material for the computer*, AEDS Twenty-first Annual Convention Proceedings, p. 10, May 1983.

8. Baughman, Jo Ann, *Developing Instruction Material for the computer*, AEDS Twenty-first Annual Convention Proceedings, p. 10, May 1983.

9. Computer-based Education Research Laboratory Report No.18, *Myth and Reality: Essential Decisions in Computer-Based Instructional design*, by E. call-Himwich and E.R. Steinberg, p. 17, 24 March 1978.

10. Computer-based Education Research Laboratory Report No.18, *Myth and Reality: Essential Decisions in Computer-Based Instructional design*, by E. call-Himwich and E.R. Steinberg, p. 11, March 1978.

11. Computer-based Education Research Laboratory Report No.18, *Myth and Reality: Essential Decisions in Computer-Based Instructional design*, by E. call-Himwich and E.R. Steinberg, p. 11, March 1978.

12. Bruce W. Morgan *An Introduction to Bayesian Statistical Decision Processes*, Prentice-Hall, p. 1, Inc. Eglewood Cliffs, New Jersey, 1968.

13. J.-A. Morales
    Bayesian Full Inference Structure Analysis, Lecture Notes in Operation Research and Mathematical Systems, p. 1, Springer-Verlag Berlin, Heidelberg, New York, 1971.

14. Robert L. Winkler *Introduction to Bayes Inference and Decision*, by Holt, Rinehart and Winston, p. 41, Inc. 1972.

15. Taracad R. Sivasankaran and Tung Bui, *A Bayesian Diagnostic Model for Intelligent CAI Systems*, Naval Postgraduate School, pp. 4-6, June 1986.

16. Taracad R. Sivasankaran and Tung Bui, *A Bayesian Diagnostic Model for Intelligent CAI Systems*, Naval Postgraduate School, pp. 6-12, June 1986.

17. William K. Kershner, *Student Pilot's Flight Manual*, Iowa State University press, pp. 6-65, Fifth edition, 1979.

18. William K. Kershner, *Student Pilot's Flight Manual*, Iowa State University press, p. 6, Fifth edition, 1979.

19. William K. Kershner, *Student Pilot's Flight Manual*, Iowa State University press, pp. 6-65, Fifth edition, 1979.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Chief of Naval Opertions Director, Information Systems(OP-945) Navy Department Washington, D.C., 20350-2000	1
4.	Department Chairman, Code 52 Department of Computer Science Naval Postgraduate School Monterey, CA 93943-5000	1
5.	Associate Professor Neil C. Rowe, Code 52Rp Department of Computer Science Naval Postgraduate School Monterey, CA 93943-5000	1
6.	Associate Professor C. Thomas Wu, Code 52Wq Department of Computer Science Naval Postgraduate School Monterey, CA 93943-5000	1
7.	Library, Royal Thai Naval Academy Smuthprakarn, Thailand	2
8.	Library, Naval Officer Colledge Royal Thai Navy, Bangkok Thailand	2
9.	Cdr. Yodchai Rugsumruad 69/1 Paholyotin km.27 Pratumtani 12130, Thailand	1
10.	Thai Navy Air Wing Sukumvith Km.12, Rayong, Thailand	2